AD 678569

GENERAL DYNAMICS
General Atomic Division

SPECIAL NUCLEAR EFFECTS LABORATORY

P.O. BOX 608, SAN DIEGO, CALIFORNIA 92112

(1)

GAMD-7310
Category A

TRIOIL
A THREE-DIMENSIONAL VERSION OF THE OIL CODE

Work done by:
W. E. Johnson

Report written by:
W. E. Johnson

This document has been approved for public release and sale; its distribution is unlimited.

This document, which was prepared primarily for internal use at General Atomic, may contain preliminary or incomplete data. It is informal and is subject to revision or correction; it does not, therefore, represent a final report.

Advanced Research Projects Agency
ARPA Order No. 71-62
Ballistic Research Laboratories
Contract No. DA-04-495-AMC-1481(X)
GA Project 6003

June 1, 1967

93

FOREWORD

The TRIOIL computer code described herein is as it existed on July 1, 1967. The code has been in continuous development for three years and in its presented form has been applied in this report. However, the development and improvement in both the physics and mathematics of the code are being continued, so that duplication of results (or even close agreement) between problems run with the code as published and the code as it existed either before or after this time is not necessarily to be expected.

General Atomic has exercised due care in preparation, but does not warrant the merchantability, accuracy, and completeness of the code or of its description contained herein. The complexity of this kind of program precludes any guarantee to that effect. Therefore, any user must make his own determination of the suitability of the code for any specific use and of the validity of the information produced by use of the code.

TABLE OF CONTENTS

1. INTRODUCTION

The TRIOIL code is a three-dimensional continuous Eulerian cartesian hydrodynamic code.* It is a natural extension of the two-dimensional OIL code (Ref. 1). The code is in an infant stage as of now but work is continuing to make it an operative tool for research in the field of numerical hydrodynamics. In this version of TRIOIL, the scalar pressure is the only stress. However, it appears to be straightforward to incorporate stresses due to strength and viscous forces as has been done for the two-dimensional version (Ref. 2) and adding two materials (Ref. 4).

The maximum size of the three-dimensional grid is limited by the present 32K to 64K storage computers. In this present version, the maximum number of cells in any single direction *IMAX for the x-direction, JMAX for the y-direction and KMAX for the z-direction) is limited to 30; further, the total number of cells is limited to 6000. But, with high speed disks and drums, as are currently being made available, one can store x-y slabs of data. This means, at any given time, we have in core memory only two slabs of x-y data, which is sufficient for the various phases of the program. This procedure will essentially remove the limitations on grid size, and enable one to apply this code to solving many types of three-dimensional problems.

A rezone routine, that increases all linear cell dimensions by a factor of two, is presently an operative feature of this code. This routine is programmed primarily for hypervelocity impact calculations into an infinite target. To use it for different applications will require some minor modifications. Eight cells in the old grid are combined into one for the new grid. Energy, mass, and momentum are conserved. This routine assumes the following:

(a) That IMAX, JMAX and KMAX are even integers. (iMAX, jMAX and kMAX are the total number of cells in the x, y, and z directions.)

*This work was funded jointly by the Systems and Research Department of Honeywell, Inc., at Minneapolis, Minnesota and General Atomic Division of General Dynamics Corporation.

(b) That the j value of the initial interface between projectile and target is 2^N, allowing N rezones.

This routine can be called for whenever mass leaves any of the six grid boundaries.

The six slides of the grid can either be reflective or transmittive. Variable zoning of Δx, Δy and Δz is a working feature of the code. Active grid counters are calculated for all three directions. By doing this, one processes only that portion of the grid that is active.

Programs for displaying density, velocities and pressures are currently being programmed using the Stromberg-Carlson 4020 Plotter.

A subroutine called SETUP is available for generating the starting data for the TRIOIL code. This routine will only generate a rectangular parallelpiped for the projectile and target. For the more complicated geometries, we plan to modify the CLAM code (Ref. 1) to generate three-dimensional starting conditions for the TRIOIL code.

An accurate estimate of computer time required to run a realistic problem in three dimensions is not presently available, however, calculations reported here, have been run on the UNIVAC-1108 and Control Data 6600 computers to a point where the peak pressures have been attenuated by several orders of magnitude for computer times of approximately one hour.

The code is written in FORTRAN IV language and is operative on the IBM-7044 and the UNIVAC-1108 as well as the Control Data 6600.

2.1 Basic Equations

The Eulerian equations in cartesian geometry are:

$$\text{(A)} \quad \frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z} = 0 \quad .$$

$$\text{(B)} \quad \rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} + \rho w \frac{\partial u}{\partial z} + \frac{\partial P}{\partial x} = 0 \quad .$$

$$\rho \frac{\partial v}{\partial t} + \rho u \frac{\partial v}{\partial x} + \rho v \frac{\partial v}{\partial y} + \rho w \frac{\partial v}{\partial z} + \frac{\partial P}{\partial y} = 0 \quad .$$

$$\rho\frac{\partial w}{\partial t}+\rho u\frac{\partial w}{\partial x}+\rho v\frac{\partial w}{\partial y}+\rho w\frac{\partial w}{\partial z}+\frac{\partial P}{\partial z}=0\ .$$

$$\text{(C)}\quad \rho\frac{\partial E}{\partial t}+\rho u\frac{\partial E}{\partial x}+\rho v\frac{\partial E}{\partial y}+\rho w\frac{\partial E}{\partial z}+\frac{\partial(Pu)}{\partial x}+\frac{\partial(Pv)}{\partial y}+\frac{\partial(Pw)}{\partial z}=0\ .$$

These equations are solved in two parts, as in the OIL code (Ref. 1) and the familiar particle-in-cell codes (Ref. 3). The transport terms on the left side of Eq. (B) and (C) are temporarily dropped, while we compute (in PH1) the momentum and energy contributions due to pressure forces only. The omitted transport contributions to the momentum and energy are later approximated when we solve equation (A) and move mass across the cell boundaries.

2.1.1. Effect of Pressure Forces Only (PH1)

Re-writing Eqs. (B) and (C) with the transport terms dropped

$$\rho\frac{\partial u}{\partial t}=-\frac{\partial P}{\partial x} \tag{1}$$

$$\rho\frac{\partial v}{\partial t}=-\frac{\partial P}{\partial y} \tag{2}$$

$$\rho\frac{\partial w}{\partial t}=-\frac{\partial P}{\partial z} \tag{3}$$

$$\rho\frac{\partial E}{\partial t}=-\frac{\partial Pu}{\partial x}-\frac{\partial Pv}{\partial y}-\frac{\partial Pw}{\partial z} \tag{4}$$

$$p=f(\rho,I)\quad \text{equation of state} \tag{5}$$

These variables, in one consistent set of units are

ρ = density of cell (L) in g/cm^3.

x = x coordinate in cm.

y = y coordinate in cm.

z = z coordinate in cm.

u = x component of velocity in cm/shake.

v = y component of velocity in cm/shake.

w = z component of velocity in cm/shake.

P = material pressure in jerks/cm^3.

E = total specific energy in jerks/g.

I = specific internal energy in jerks/g

t = time in shakes

1 shake = 10^{-8} sec

1 jerk = 10^{16} ergs

There are no built-in units for this code, the user can select his own units in a given application by way of the equation of state constants and input data.

The density, pressure, velocities and internal energy are all cell-centered quantities referred to with the index (L).

Rewriting Eq. 4:

$$\rho \frac{\partial E}{\partial t} = - \frac{\partial Pu}{\partial x} - \frac{\partial Pv}{\partial y} - \frac{\partial Pw}{\partial z}$$

or

$$\rho \frac{\partial}{\partial t} \left[I + \tfrac{1}{2}(u^2+v^2+w^2) \right] = - \frac{\partial Pu}{\partial x} - \frac{\partial Pv}{\partial y} - \frac{\partial Pw}{\partial z}$$

$$\rho \frac{\partial I}{\partial t} + \rho u \frac{\partial u}{\partial t} + \rho v \frac{\partial v}{\partial t} + \rho w \frac{\partial w}{\partial t} = - P \frac{\partial u}{\partial x} - u \frac{\partial P}{\partial x} - P \frac{\partial v}{\partial y} - v \frac{\partial P}{\partial y} - P \frac{\partial w}{\partial z} - w \frac{\partial P}{\partial z}$$

but

$$\rho \frac{\partial u}{\partial t} = - \frac{\partial P}{\partial x} \quad \text{and} \quad \rho \frac{\partial v}{\partial t} = - \frac{\partial P}{\partial y} \quad \text{and} \quad \rho \frac{\partial w}{\partial t} = - \frac{\partial P}{\partial z}$$

thus

$$\rho \frac{\partial I}{\partial t} = - P \left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right] \qquad (6)$$

Now we can write the four differential equations (1, 2, 3 and 6) in difference form.

The storage arrays for the cell centered quantities (mass, velocities, pressure and specific internal energy are as follows (Fig. 1).

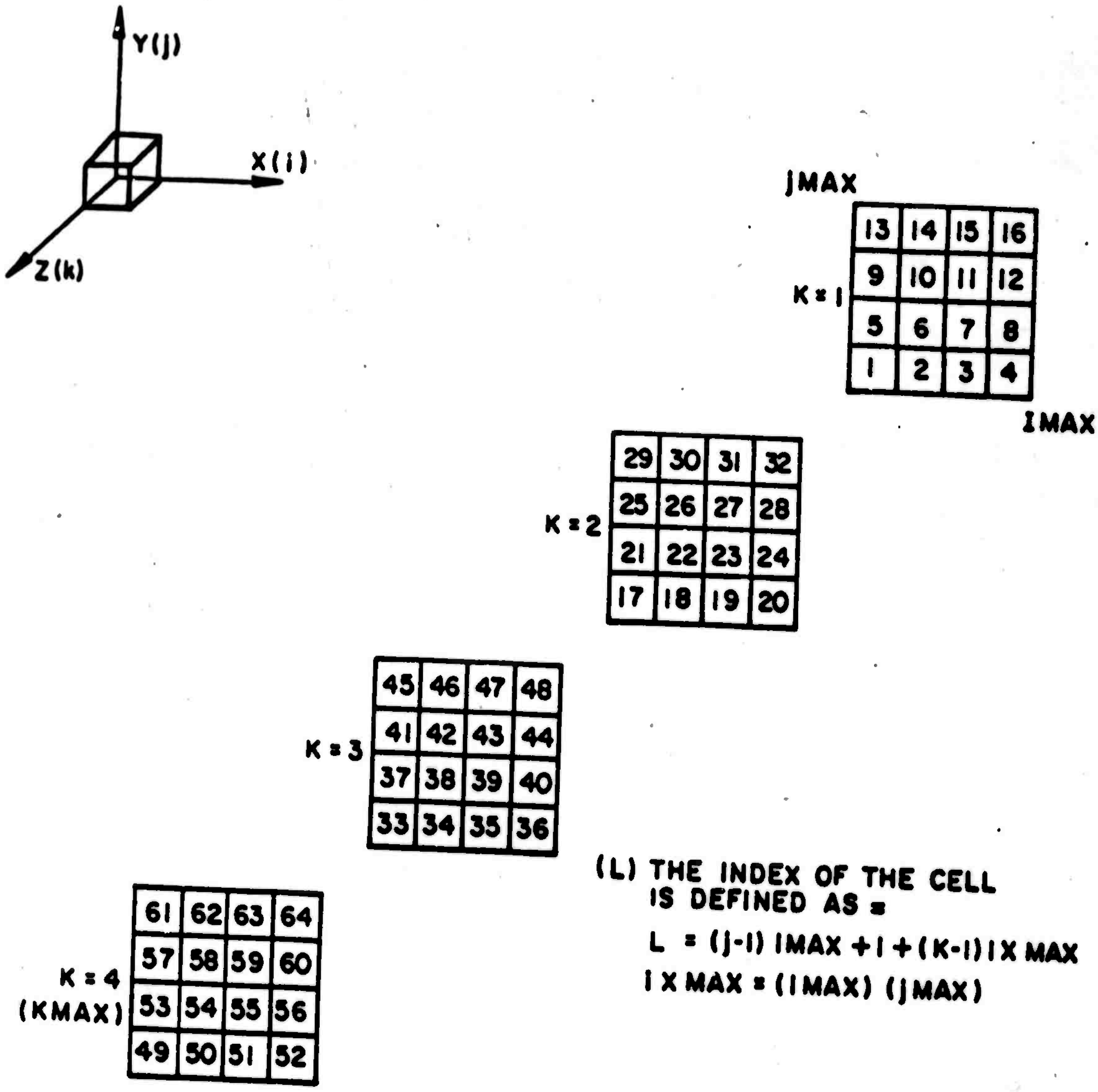

Y(j)
X(i)
Z(k)
JMAX
13 14 15 16
9 10 11 12
5 6 7 8
1 2 3 4
K = 1
IMAX
29 30 31 32
25 26 27 28
21 22 23 24
17 18 19 20
K = 2
45 46 47 48
41 42 43 44
37 38 39 40
33 34 35 36
K = 3
61 62 63 64
57 58 59 60
53 54 55 56
49 50 51 52
K = 4
(KMAX)
(L) THE INDEX OF THE CELL IS DEFINED AS =
L = (j-1) IMAX + I + (K-1) IX MAX
I X MAX = (IMAX) (JMAX)

Fig. 1

In the discussion to follow, please refer to Fig. 2. Re-writing the equations (1 to 3) in difference form results in:

The x-momentum equation (1) becomes in difference form

$$\rho_L^n \left(\frac{\tilde{u}_L - u_L^n}{\Delta t} \right) = \frac{P_{L-1}^n - P_{L+1}^n}{2\Delta x(i)}$$

The y-momentum equation (2) becomes in difference form

$$\rho_L^n \left(\frac{\tilde{v}_L - v_L^n}{\Delta t} \right) = \frac{P_{L-iMAX}^n - P_{L+iMAX}^n}{2\Delta y(j)}$$

The z-momentum equation (3) becomes in difference form

$$\rho_L^n \left(\frac{\tilde{w}_L - w_L^n}{\Delta t} \right) = \frac{P_{L-(iMAX)(jMAX)}^n - P_{L+(iMAX)(jMAX)}^n}{2\Delta z(k)}$$

Here the acceleration of cell L is seen to depend only on the pressures in the neighbor cells (not that of L). Defining pressures at interfaces,

$$PL^n = \frac{P_{L-1}^n + P_L^n}{2.}$$

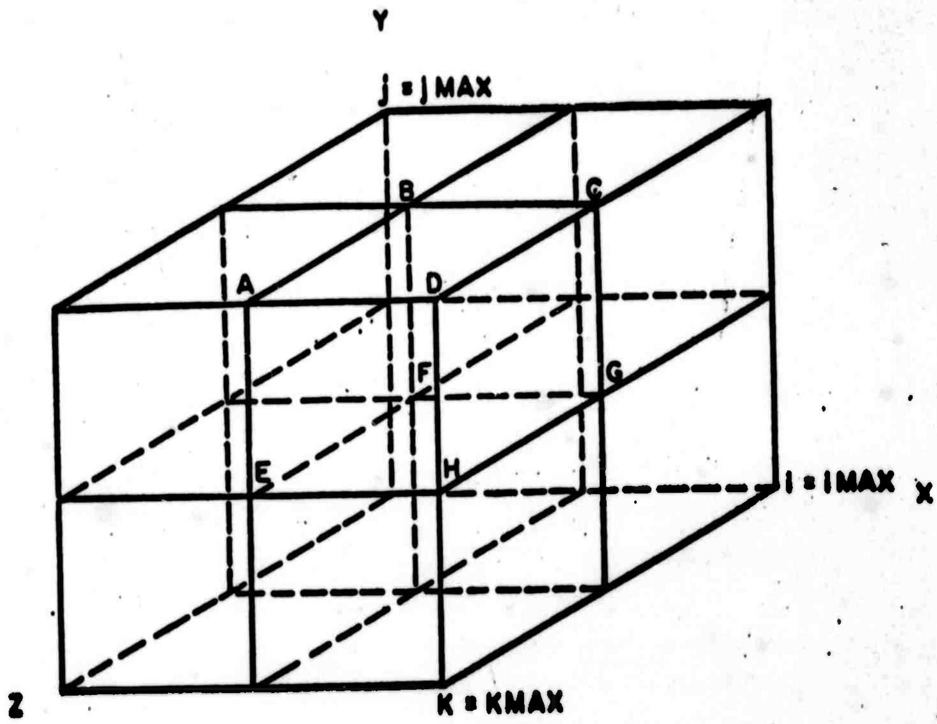
$$PRR^n = \frac{P_L^n + P_{L+1}^n}{2.}$$

$$PBLO^n = \frac{P_{L-iMAX}^n + P_L^n}{2.}$$

$$PABOVE^n = \frac{P_L^n + P_{L+iMAX}^n}{2.}$$

$$PBIND^n = \frac{P_{L-(iMAX)(jMAX)}^n + P_L^n}{2.}$$

$$PZR^n = \frac{P_{L+(iMAX)(jMAX)}^n + P_L^n}{2.}$$



(L) THE INDEX OF THE CELL
IS DEFINED AS =

$$L = (j-1)\,IMAX + i + (K-1)\,IX\,MAX$$

$$IX\,MAX = (i\,MAX)(j\,MAX)$$

$$X(i) = \sum_{i=1}^{i} \Delta X(i)$$

$$Y(j) = \sum_{j=1}^{j} \Delta Y(j)$$

$$Z(K) = \sum_{K=1}^{K} \Delta Z(K)$$

Fig. 2

Substituting these interface pressures into the three-momentum equations results in:

$$\tilde{u}_L - u_L^n = \frac{\Delta t}{\rho_L^n}\left[\frac{PL^n - PRR^n}{\Delta x(i)}\right]$$

$$\tilde{v}_L - v_L^n = \frac{\Delta t}{\rho_L^n}\left[\frac{PBLO^n - PABOVE^n}{\Delta y(j)}\right]$$

$$\tilde{w}_L - w_L^n = \frac{\Delta t}{\rho_L^n}\left[\frac{PBIND^n - PZR^n}{\Delta z(j)}\right]$$

where the usual ~ (tilda) designates the new velocities (not at cycle n+1), since we have temporarily dropped the transport terms).

The specific internal energy equation becomes

$$\rho_L \frac{\tilde{I}_L - I_L^n}{\Delta t} = - P_L^n \left[\frac{v_{L-iMAX}^{n+\frac{1}{2}} - v_{L+iMAX}^{n+\frac{1}{2}}}{2\Delta y(j)} + \frac{u_{L-1}^{n+\frac{1}{2}} - u_{L+1}^{n+\frac{1}{2}}}{2\Delta x(i)} + \frac{w_{L-(iMAX)(jMAX)}^{n+\frac{1}{2}} - w_{L+(iMAX)(jMAX)}^{n+\frac{1}{2}}}{2\Delta z(k)}\right]$$

The reason for the velocities at time $(n+\frac{1}{2})$ can be seen from energy conservation considerations, which we will discuss later in the text. Defining

$$u_L^{n+\frac{1}{2}} = \frac{u_L^n + \tilde{u}_L}{2.}$$

$$v_L^{n+\frac{1}{2}} = \frac{v_L^n + \tilde{v}_L}{2.}$$

$$w_L^{n+\frac{1}{2}} = \frac{w_L^n + \tilde{w}_L}{2.}$$

and

$$\text{VBLO} = \frac{v_L + v_{L-iMAX}}{2.}$$

$$\text{VABOVE} = \frac{v_L + v_{L+iMAX}}{2.}$$

$$\text{UL} = \frac{U_L + U_{L-1}}{2.}$$

$$\text{URR} = \frac{U_L + U_{L+1}}{2.}$$

$$\text{UBIND} = \frac{W_L + W_{L-(iMAX)(jMAX)}}{2.}$$

$$\text{WZR} = \frac{W_L + W_{L+(iMAX)(jMAX)}}{2.}$$

then

$$\rho_L^n \left(\frac{\tilde{I}_L - I_L^n}{\Delta t} \right) = P_L^n \left[\frac{\text{VBLO}^n + \widetilde{\text{VBLO}}}{2\Delta y(j)} - \frac{\text{VABOVE}^n + \widetilde{\text{VABOVE}}}{2\Delta y(j)} + \frac{\text{UL}^n + \widetilde{\text{UL}}}{2\Delta X(i)} - \frac{\text{URR}^n + \widetilde{\text{URR}}}{2\Delta X(i)} + \frac{\text{UBIND}^n + \widetilde{\text{UBIND}}}{2\Delta Z(k)} - \frac{\text{WZR}^n + \widetilde{\text{WZR}}}{2\Delta Z(k)} \right] .$$

The solution of the momentum equations are very straightforward, however, the solution to the energy equation requires the velocities at two different time steps (the old and new velocities). We have chosen to make two passes through the routine, the first pass to integrate the momentum equations, but formulate the interface velocities first for the work term contribution to the internal energy before integrating the momentum equations (since we have allowed only one array per velocity component).

The second pass, we bypass the momentum equations, and only compute the new interface velocities for their contribution to the work term. A single pass can be done by looking ahead two cells above; two cells to the right and two cells in front.

The choice of the velocities at $n + \frac{1}{2}$ in the energy equation is apparent in the following discussion. For convenience we will go through the logic in the y direction. Since we have dropped the transport terms, our integration of the momentum and energy equations have not been advanced to time $(n + 1)$. As before, we designate the phase 1 velocities and energy as $\tilde{\mu}$, $\tilde{v}$, w, and $\tilde{I}$.

$$\tilde{v}_{j-\frac{1}{2}} = v^n_{j-\frac{1}{2}} + \frac{\Delta t}{\rho^n_{j-\frac{1}{2}}} \left[\frac{P^n_{j-3/2} - P^n_{j+\frac{1}{2}}}{2\Delta y(j)} \right]$$

and

$$\tilde{I}_{j-\frac{1}{2}} = I^n_{j-\frac{1}{2}} + \frac{\Delta t \, P^n_{j-\frac{1}{2}}}{\rho^n_{j-\frac{1}{2}}} \left[\frac{\bar{v}_{j-3/2} - \bar{v}_{j+\frac{1}{2}}}{2\Delta y(j)} \right]$$

where

$$\bar{v}_{j-3/2} = \frac{\tilde{v}_{j-3/2} + v^n_{j-3/2}}{2.}$$

and

$$\bar{v}_{j+\frac{1}{2}} = \frac{\tilde{v}_{j+\frac{1}{2}} + v^n_{j+\frac{1}{2}}}{2.}$$

Before entering PH1, where the quantities are at time n, the total energy of the system (considering one dimension, the y direction) is:

$$E^n = \sum_{j=1}^{JMAX} AMX^n_{j-\frac{1}{2}} \left[I^n_{j-\frac{1}{2}} + \frac{1}{2} \left(v^n_{j-\frac{1}{2}} \right)^2 \right] .$$

And the total energy at the end of PH1 is then

$$\tilde{E} = \sum_{j=1}^{JMAX} AMX^n_{j-\frac{1}{2}} \left[\tilde{I}_{j-\frac{1}{2}} + \frac{1}{2} \left(\tilde{v}_{j-\frac{1}{2}} \right)^2 \right] .$$

The change, $\Delta E = E^n - \tilde{E}$ should be zero for energy conservation or

$$\Delta E = \sum_{j=1}^{JMAX} AMX^n_{j-\frac{1}{2}} \left[I^n_{j-\frac{1}{2}} - \tilde{I}_{j-\frac{1}{2}} + \frac{1}{2} \left(v^n_{j-\frac{1}{2}} \right)^2 - \frac{1}{2} \left(\tilde{v}_{j-\frac{1}{2}} \right)^2 \right]$$

the Δ kinetic terms can be represented by

$$\frac{\left(v^n_{j-\frac{1}{2}} + \tilde{v}_{j-\frac{1}{2}}\right)}{2.} \left[v^n_{j-\frac{1}{2}} - \tilde{v}_{j-\frac{1}{2}}\right]$$

$$\Delta E = \sum_{j=1}^{JMAX} AMX^n_{j-\frac{1}{2}} \left[I^n_{j-\frac{1}{2}} - \tilde{I}_{j-\frac{1}{2}} + \bar{v}_{j-\frac{1}{2}} \left(v^n_{j-\frac{1}{2}} - \tilde{v}_{j-\frac{1}{2}}\right)\right]$$

$$= \sum_{j=1}^{JMAX} AMX^n_{j-\frac{1}{2}} \left[- \frac{\Delta t \, P^n_{j-\frac{1}{2}}}{\rho^n_{j-\frac{1}{2}}} \left(\frac{\bar{v}_{j-3/2} - \bar{v}_{j+\frac{1}{2}}}{2 \, \Delta y_j}\right)\right.$$

$$\left. - \bar{v}_{j-\frac{1}{2}} \frac{\Delta t}{\rho^n_{j-\frac{1}{2}}} \left(\frac{P^n_{j-3/2} - P^n_{j+\frac{1}{2}}}{2 \, \Delta y_j}\right)\right]$$

$$= \Delta t \sum_{j=1}^{JMAX} \frac{AMX^n_{j-\frac{1}{2}}}{\rho^n_{j-\frac{1}{2}} \, 2 \, \Delta y_j} \left[- P^n_{j-\frac{1}{2}} \bar{v}_{j-3/2}\right.$$

$$\left. + P^n_{j-\frac{1}{2}} \bar{v}_{j+\frac{1}{2}} - P^n_{j-3} \bar{v}_{j-\frac{1}{2}} + P^n_{j+\frac{1}{2}} \bar{v}_{j-\frac{1}{2}}\right]$$

$$= \frac{-\Delta t}{2} DX(i) DZ(k) \sum_{j=1}^{JMAX} \left[P^n_{j-\frac{1}{2}} \bar{v}_{j-3/2} + P^n_{j-3/2} \bar{v}_{j-\frac{1}{2}}\right.$$

$$\left. -P^n_{j+\frac{1}{2}} \bar{v}_{j-\frac{1}{2}} - P^n_{j-\frac{1}{2}} \bar{v}_{j+\frac{1}{2}}\right]$$

Thus, the last two terms in j being cancelled by the first two terms in $j+1$. By prescribing the proper boundary conditions, we will have energy conservation for the entire grid.

EXAMPLE:

for $j = 1$

$$P_{\frac{1}{2}} v_{-\frac{1}{2}} + P_{-\frac{1}{2}} v_{\frac{1}{2}} - P_{3/2} v_{\frac{1}{2}} - P_{\frac{1}{2}} v_{3/2}$$

$j = 2$

$$P_{3/2} v_{\frac{1}{2}} + P_{\frac{1}{2}} v_{3/2} - P_{5/2} v_{3/2} - P_{3/2} v_{5/2}$$

$j = 3$

$$P_{5/2} v_{3/2} + P_{3/2} v_{5/2} - P_{7/2} v_{5/2} - P_{5/2} v_{7/2}$$

Note, the boundary terms do not cancel. For the example, assume the bottom boundary to be reflective. Referring to Eq. (A), the first two terms do not cancel. If, however we set the pressure of the mirror cell $P^n_{-\frac{1}{2}} = P^n_{\frac{1}{2}}$ (this does not imply that $v_{\frac{1}{2}} = 0$) and $v_{-\frac{1}{2}} = -v_{\frac{1}{2}}$, this does lead to cancellation of the first two terms. If, however, we designate the bottom boundary to be transmittive, our boundary conditions are $\dot{v}_{\frac{1}{2}} = 0_j$ which implies $P^n_{-\frac{1}{2}} = P^n_{3/2}$, and that $v_{-\frac{1}{2}} = v_{\frac{1}{2}}$. This still leaves us with the two terms $P_{\frac{1}{2}} v_{\frac{1}{2}} + P_{3/2} v_{\frac{1}{2}}$, thus we compensate for these terms, by adding or subtracting energy to the system. To keep all the books straight, E th must be modified accordingly. Similar boundary conditions may exist for the other four sides of the grid.

2.1.2 Adding the Transport Terms (PH2)

Rewriting Eq. (A) the mass transport equation in finite difference form results in

$$\frac{\rho_L^{n+1} - \rho_L^{n}}{\Delta t} = \left[\frac{\rho^n_{j-1} \tilde{v}_{j-1} - \rho^n_j \tilde{v}_j}{\Delta y(j)} + \frac{\rho^n_{i-1} \tilde{u}_{i-1} - \rho_i \tilde{u}_i}{\Delta X_{(i)}} + \frac{\rho^n_{k-1} \tilde{w}_{k-1} - \rho^n_k \tilde{w}_k}{\Delta Z(k)} \right]$$

or in the formulation for mass we have

$$\Delta M_L = \Delta t \left[(Av)^y_B \rho - (Av)^y_T \rho + (Au)^X_L \rho - (Au)^X_R \rho + (Aw)^Z_{BH} \rho - (Aw)^Z_F \rho \right]$$

where the superscripts X, Y, Z, refer to the particular coordinate, and B = bottom, T = top, L = left, R = right, BH = behind and F = front. The mass that is moved from cell to cell, has then momentum and energy associated with it, thus these are the approximations to the transport terms that were omitted in PH1 for the momentum and energy equations.

The ρ (density) is that of the donor cell, and the velocity is yet to be determined. Various techniques for the velocity weighting have been tried (Ref. 1). The velocity weighting scheme in this report is identical to that in the OIL report (Ref. 1).

2.1.3 Time Control

The time step (Δt) is controlled by the Courant condition and the condition that the mass flux equation will not empty more mass from a cell than is there.

Take the y direction (a similar treatment is done for the other two directions)

$$\Delta M_y = \rho \bar{v} A \Delta t$$

Let

$$\bar{v} = v_{(L)}$$

$$\rho = \rho_{(L)}$$

$$\Delta M_y = AMX_{(L)}$$

then

$$AMX(L) = \rho_{(L)} \, v_{(L)} \, A_j \, \Delta t$$

$$= \frac{AMX_{(L)}}{DX(i)\, Dy(j)\, DZ(k)} \, v_{(L)} \, DX(i)\, DZ(k)\, \Delta t$$

$$= \frac{AMX_{(L)} \, v_{(L)} \, \Delta t}{Dy(j)}$$

or $|v_{(L)}|\ \Delta t \le Dy(j)$ such that the flux in the y direction will not empty the cell (L).

The Courant condition is that $\frac{C}{C_M} \le 1$ where C = sound speed and C_M is the maximum speed at which a disturbance can propagate in the given grid or $C\frac{\Delta t}{\Delta x} \le 1.$, or $C\frac{\Delta t}{\Delta y} \le 1.$, and $C\frac{\Delta t}{\Delta z} \le 1.$

2.1.4 Remarks

No corner coupling (that is, mass is constrained to move at right angles to the sides of the cell) exists in this version, and no attempt to systematically study this has been initiated. The movement of mass across the cell boundaries give rise to forces which are effective in reducing fluctuations that arise from the differencing technique. That is of the form of a "true" viscosity, being proportional to the velocity gradient. It is this force which enables the Eulerian codes to treat shocks, where again, as in the Q method used in Lagrangian codes, the shock is spread over two or three zones.

2.2 Logic of TRIOIL

The logic involved in following a given cell (L) from t to $t + \Delta t$ or cycle n to $n + 1$ is as follows:

We assume we have integrated the mass, the three velocity components and the internal energy to cycle n, now all that remains to complete cycle n, and to begin cycle $n + 1$, is to update the pressures from the equation of state and calculate a new time step.

2.2.1 CDT Routine

Here we calculate the pressure (P) array for the entire grid. The pressure $(P_L) = f(\rho_L, I_L)$ where L is the index of the cell in question, defined as $L = (j-1)\,iMAX + i + (k-1)\,(iMAX)\,(jMAX)$. The density $\left(\rho_L = \frac{AMX(L)}{DX(i)\ Dy(j)\ DZ(k)}\right)$ is not one of the variables, it must be calculated several times during the cycle. It is planned in the future, to replace the mass storage with density. The speed of sound $C = (\frac{\gamma P}{\rho})^{1/2}$ for a polytropic equation of state) or $(\frac{\partial P}{\partial \rho})_s^{1/2}$ for a general form is then calculated. From the particle velocities and speed of sound, a new Δt is calculated.

(Options for Δt control are identical to that in OIL.) The cycle number and the time are now advanced.

2.2.2 The EDIT Routine

This code has three separate editing routines all included in the subroutine EDIT. The first of these is the routine called "Short Print". This displays the problem number, cycle number, time, internal and kinetic energy, energy check and indices of the cell that is controlling the time step. The next routine available is called "Long Print". Here one edits each column, the three velocity components, pressure, mass, density, specific internal energy and y, all versus j (the index of the rows). Thus there are kMAX sets of these, beginning with k=1. The normal units chosen are gram-cm-sh. which gives a logical unit for the pressure as 10^{16} ergs/cm^3 and for the specific internal energy as 10^{16} ergs/gram.

2.2.3 PH1 Routine

Here we integrate the three momentum equations and the internal energy equation due to pressure forces only. No material is moved at this time, and the transport terms are dropped temporarily. Using the new pressure and Δt, we can solve the momentum and energy equations.

PL(j), the pressure at interface (i-1) and UL(j), the velocity at interface (i-1) are available from the previous column sweep on i-1:

$$PL(j) = \frac{P_L^n + P_{L-1}^n}{2}$$

$$UL(j) = \frac{U_L^n + U_{L-1}^n}{2.}$$

The PBLO term which was the PABOVE for the cell below (L-iMAX) and VBLO which was VABOVE for cell below, are also available for interface j-1.

$$PBLO = \frac{P_L^n + P_{L-iMAX}^n}{2.}$$

$$VBLO = \frac{V_L^n + V_{L-iMAX}^n}{2.}$$

The PBIND term (the pressure at the back surface of the cell) which was PZR for cell L - (iMAX) (jMAX) and UBIND (the velocity at the back surface of the cell) which was WZR for cell L - (iMAX) (jMAX) are also available for interface K-1.

$$PBIND = \frac{P_L^n + P_{L-(iMAX)(jMAX)}^n}{2.}$$

$$UBIND = \frac{W_L^n + W_{L-(iMAX)(jMAX)}^n}{2.}$$

Thus, we need only to calculate quantities at the top, the right, and the front of cell (L). At the top we calculate

$$PABOVE = \frac{P_L^n + P_{L+iMAX}^n}{2.}$$

and VABOVE $= \dfrac{V_L^n + V_{L+iMAX}^n}{2.}$; at the right we calculate PR $= \dfrac{P_L^n + P_{L+1}^n}{2.}$

and URR as $= \dfrac{U_L^n + U_{L+1}^n}{2.}$ and in front as

$$PZR = \frac{P_L^n + P_{L+(iMAX)(jMAX)}^n}{2.}$$

$$WZR = \frac{W_L^n + W_{L+(iMAX)(jMAX)}^n}{2.}$$

When the cell in question is void, the pressures at the top, right and front interface are set to zero and the velocities are set = to the velocity of the cell above, to the right, and in front respectively. If a occupied cell has a void neighbor, the pressure at that interface is set = 0, and the velocity at that interface is set = to the velocity of the occupied cell in question.

We now have sufficient information to integrate the three momentum equations and part of the internal energy equation.

$$\rho \frac{\partial u}{\partial t} = \frac{-\partial p}{\partial x}$$

or

$$\tilde{U}_L = U_L^n + \frac{PL_{(j)}^n - PRR^n}{AMX_L^n} Dy(j)\ DZ(k)\ \Delta t$$

and

$$\rho \frac{\partial v}{\partial t} = \frac{-\partial p}{\partial y}$$

or

$$\tilde{v}_L = v_L^n + \frac{PBLO^n - PABOVE^n}{AMX_L^n} DX(i)\ DZ(k)\ \Delta t$$

and

$$\rho \frac{\partial w}{\partial t} = \frac{-\partial p}{\partial z}$$

or

$$\tilde{W}_L = W_L^n + \frac{\left(PBIND(M)^n - PZR^n\right)}{AMX_L^n} Dy(j)\ DX(i)\ \Delta t$$

where the index m refers to the (i,j) value of the slab x-y just behind. We can add the work terms due to velocities at cycle n to the change in internal energy as

$$\tilde{I}_L^{(1)} = I_L^n + \frac{P_L^n\ \Delta t}{AMX_L^n} \left[\frac{(VBLO^n - VABOVE^n)}{2.} DX(i)\ DZ(k) + \frac{\left(UL(j)^n - URR^n\right)}{2.} Dy(j)\ DZ(k) + \frac{\left(UBIND(m)^n - WZR^n\right)}{2.} DX(i)\ Dy(j)\right]$$

Now, one more pass is made through the entire grid, this time omitting the momentum equations but calculating the interface velocities, resulting in the integration of the internal energy to time (~).

$$\tilde{I}_L = \tilde{I}_L^{(1)} + \frac{P_L^n\ \Delta t}{AMX_L^n} \left[\frac{(V\tilde{B}LO - VAB\tilde{O}VE)}{2.} DX(i)\ DZ(k) + \frac{\left(U\tilde{L}(j) - U\tilde{R}R\right)}{2.} Dy(j)\ DZ(k) + \frac{\left(UB\tilde{I}ND(m) - W\tilde{Z}R\right)}{2.} DX(i)\ Dy(j)\right]$$

The option to integrate backwards from time (~) to n if a negative internal energy is encountered, is not available in this version.

2.2.4 PH2 Routine

Here, we move mass across the fixed boundaries. Momenta and energy is carried across with this mass and this approximates the transport terms omitted from the momentum and energy equations in PH1. Please refer to Fig. 2 for the following discussion.

The points A, B, C, D, E, F, G, and H are the eight corners of the cell. The following notation will be followed: side AEFB refers to the left, side ABCD refers to the top, CDHG refers to the right, BFCG to behind, EFGH to bottom, and ADEH to the front.

The five quantities associated with each interface are as follows:

TOP	AMPY	= mass crossing the top
	AMUT	= X momentum of this mass
	AMVT	= Y momentum of this mass
	AMWT	= Z momentum of this mass
	DELET	= specific energy across the top
RIGHT	AMMP	= mass crossing the right
	AMUR	= X momentum of this mass
	AMVR	= Y momentum of this mass
	AMWR	= Z momentum of this mass
	DELER	= specific energy across the right
BOTTOM	AMMY	= mass crossing the right
	AMMU	= X momentum of this mass
	AMMV	= Y momentum of this mass
	AMMW	= Z momentum of this mass
	DELEB	= specific energy across the bottom
LEFT	GAMC (j)	= mass crossing the left
	FLEFT (j)	= X momentum of this mass
	YAMC (j)	= Y momentum of this mass
	ZMOM (j)	= Z momentum of this mass
	SIGC (j)	= specific energy across the left
BEHIND	BMASS (M)	= mass crossing the back surface
	BXMOM (M)	= X momentum of this mass
	BYMOM (M)	= Y momentum of this mass
	BZMOM (M)	= Z momentum of this mass
	BENR (M)	= specific energy across the back surface

FRONT	FMASS	= mass crossing the front
	FXMOM	= X momentum of this mass
	FYMOM	= Y momentum of this mass
	FZMOM	= Z momentum of this mass
	FENR	= specific energy across the front

Following the typical cell (L) the masses, the momentas, and the specific energies are now available at the left and bottom and back boundaries of cell (L) from the previous column sweep, the cell below and the previous X-Y slab (K-1).

The proper boundary conditions are first set for the cell (L=1). We then begin by calculating

$$\text{VABOVE} = \frac{\tilde{V}_L + \tilde{V}_{L+iMAX}}{2.} \quad \text{for the top}$$

$$\text{URR} = \frac{\tilde{U}_L + \tilde{U}_{L+1}}{2.} \quad \text{for the right}$$

$$\text{WOUT} = \frac{\tilde{W}_L + \tilde{W}_{L+(iMAX)(jMAX)}}{2.} \quad \text{for in front}$$

Then form

$$\text{VABOVE} = \frac{\text{VABOVE}}{1 + \frac{(\tilde{V}_{L+iMAX} - \tilde{V}_L)}{\Delta y(j)} \Delta t}$$

$$\text{URR} = \frac{\text{URR}}{1 + \frac{(\tilde{U}_{L+1} - \tilde{U}_L)}{\Delta x(i)} \Delta t}$$

and

$$\text{WOUT} = \frac{\text{WOUT}}{1 + \frac{(\tilde{W}_{L+(iMAX)(jMAX)} - \tilde{W}_L)}{\Delta z(k)} \Delta t}$$

Now we can calculate the mass crossing the three boundaries as

$$\Delta M_{TOP} = \text{AMPY} = \frac{\text{AMX}(M)^n \, \text{VABOVE} \, \Delta t}{\Delta y(j)}$$

$$\Delta M_{RIGHT} = \text{AMMP} = \frac{\text{AMX}(M)^n \, \text{URR} \, \Delta t}{\Delta x(i)}$$

$$AM_{FRONT} = FMASS = \frac{AMX(M)^n\ WOUT\ \Delta t}{\Delta z(k)}$$

where (M) is the index of the donor cell. The donor cell is calculated from the sign of the weighted velocity.

The momenta associated with these three masses are now calculated where the velocity in the momenta is from the donor cell. The total specific energy that these mass fluxes carry are calculated at this time also. The momenta associated with the flux at the top is:

$$X \text{ component} = AMUT = AMPY\ [\tilde{U}(N)]$$

$$Y \text{ component} = AMVT = AMPY\ [\tilde{V}(N)]$$

$$Z \text{ component} = AMWT = AMPY\ [\tilde{W}(N)]$$

$$\text{Specific energy} = DELET = \tilde{I}_{(N)} + \frac{[\tilde{U}^2_{(N)} + \tilde{V}^2_{(N)} + \tilde{W}^2_{(N)}]}{2.}$$

Those associated with the flux at the right are:

$$X \text{ component} = AMUR = AMMP\ [\tilde{U}(N)]$$

$$Y \text{ component} = AMVR = AMMP\ [\tilde{V}(N)]$$

$$Z \text{ component} = AMWR = AMMP\ [\tilde{W}(N)]$$

$$\text{Specific energy} = DELER = \tilde{I}_{(N)} + \frac{[\tilde{U}^2_{(N)} + \tilde{V}^2_{(N)} + \tilde{W}^2_{(N)}]}{2.}$$

Those associated with the flux in front are:

$$X \text{ component} = FMASS\ [\tilde{U}(N)] = FXMOM$$

$$Y \text{ component} = FMASS\ [\tilde{V}(N)] = FYMOM$$

$$Z \text{ component} = FMASS\ [\tilde{W}(N)] = FZMOM$$

$$\text{Specific energy} = \tilde{I}_{(N)} + \frac{[\tilde{U}^2_{(N)} + \tilde{V}^2_{(N)} + \tilde{W}^2_{(N)}]}{2.} = FENR$$

Where again (N) = index of the donor cell.

The mass now in cell (L) is equal to DELM = AMX(L) + GAMC (j) + AMMY - AMPY - AMMP + BMASS M) - FMASS which equals the original mass plus the mass flow across the left, the bottom, behind, less the mass flow across the top, right and in front.

The total X momenta that has come into or left cell (L) is = SIGMU = Fleft(j) + AMMU +BXMOM(M) - AMUT - AMUR - FXMOM = the momenta crossing the left boundary plus the momenta crossing the bottom boundary plus the momenta crossing the back boundary less the momenta crossing the top, the right and front boundary.

The total Y momenta that has come into or left cell (L) = SIGMV = YAMC(j) + AMMV + BYMOM(M) - AMVT - AMVR - FYMOM = momenta crossing the left, the bottom and behind less the momenta crossing the top, right and front boundary.

The total Z momenta that has come into or left cell (L) is = SIGMW = ZMOM(j) + AMMW + BZMOM(M) - AMWT - AMWR - FZMOM = momenta crossing the left, the bottom and behind less the momenta crossing the top, right and front boundary.

Now by conserving momenta and total energy, we can calculate new velocities and specific internal energy of cell (L)

$$MU_{LE} + MU_B + MU_{BH} - MU_T - MU_R - MU_F + MU_{(L)} = (DELM)\ U^{n+1}_{(L)}$$

and

$$MV_{LE} + MV_B + MV_{BH} - MV_T - MV_R - MV_F + MV_{(L)} = (DELM)\ V^{n+1}_{(L)}$$

and

$$MW_{LE} + MW_B + MW_{BH} - MW_T - MW_R - MW_F + MW_{(L)} = (DELM)\ W^{n+1}_{(L)}$$

and solve for the three velocities at cycle n+1.

The new specific internal energy is the total less the kinetic =

$$I^{n+1}_{(L)} = \frac{E_{LE} + E_B + E_{BH} - E_T - E_R - E_F + E_L}{DELM} - \frac{(U^{n+1}_L)^2 + (V^{n+1}_L)^2 + (W^{n+1}_L)^2}{2.}$$

DELM = the new mass. The subscripts LE, B, BH, T, R, F, L, refer to the left, bottom, behind, top, right, front and cell in question.

Now the five variables that were calculated for the top of cell (L) become the bottom quantities for cell (L + iMAX) and the variables that were calculated for the right of cell (L) become the left quantities for cell (L+1) and the variables that were calculated for the front of cell (L) become the back quantities for cell L + (iMAX)(jMAX). This completes the calculation for cell (L). After completion of PH2, all that remains to complete cycle n+1 is to update the pressures, which is done in the CDT routine.

3. TEST PROBLEMS

A series of test problems were undertaken to check out the TRIOIL code.

The early test problem consisted of 8 cells (2 in the x, 2 in the y and 2 in the z direction) containing internal energy only, free to expand into a vacuum. The purpose was to check the free surface treatment, the symmetry of disturbance in the three directions and the possible sphericity of the expansion at large distances. The results were encouraging, exact symmetry existed and the expansion was spherical considering the coarse zoning.

The second test problem (as shown in Fig. 3) consists of a grid 11 x 11 x 11. The corner cell has 10^{16} ergs/g, surrounded by like material but cold. The three adjacent sides were treated as reflective boundaries, while the three opposite sides were transmittive boundaries. A check of computer results were made against the G. I. Taylor strong shock-point source solution. The comparison is presented in Fig. 4. As indicated, the position of the shock front is in good agreement with theory, however the magnitude is somewhat less in the TRIOIL solution. The latter results are consistent with similar calculations performed with the OIL (Ref. 1) code.

The third test problem was a normal impact to be compared with the OIL code. The configuration (as indicated in Fig. 5) for the TRIOIL code was a cube of 4 x 4 x 4 cells impacting normal to a semi-infinite target.

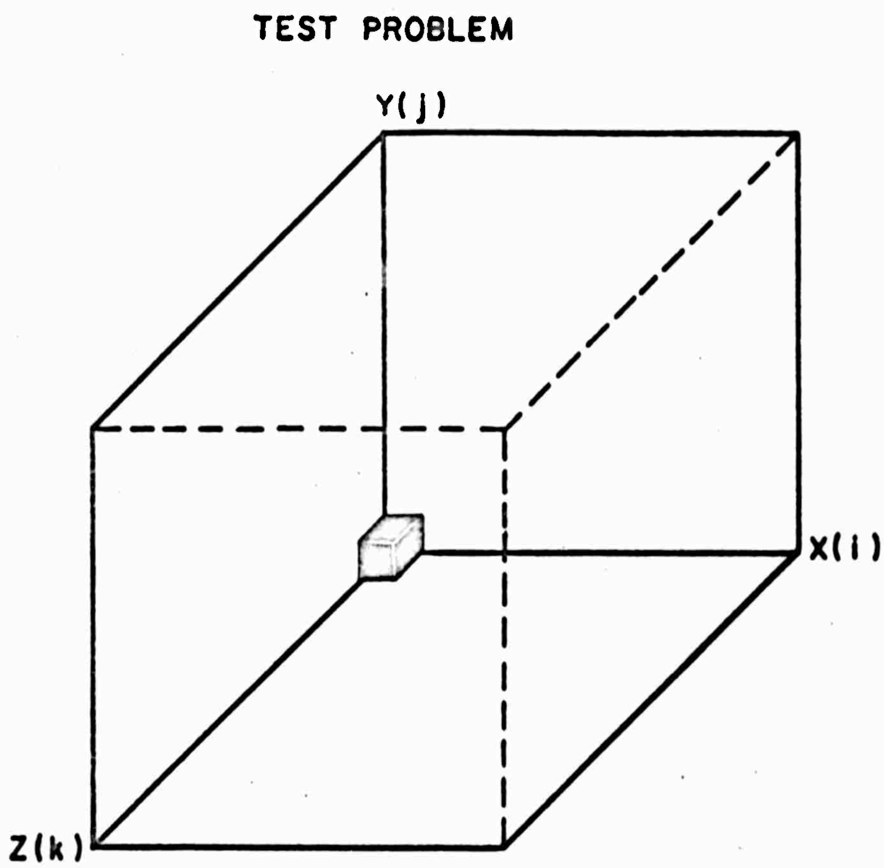


Fig. 3--The initial configuration for a point source calculation using the 3-D code for a 1-D problem

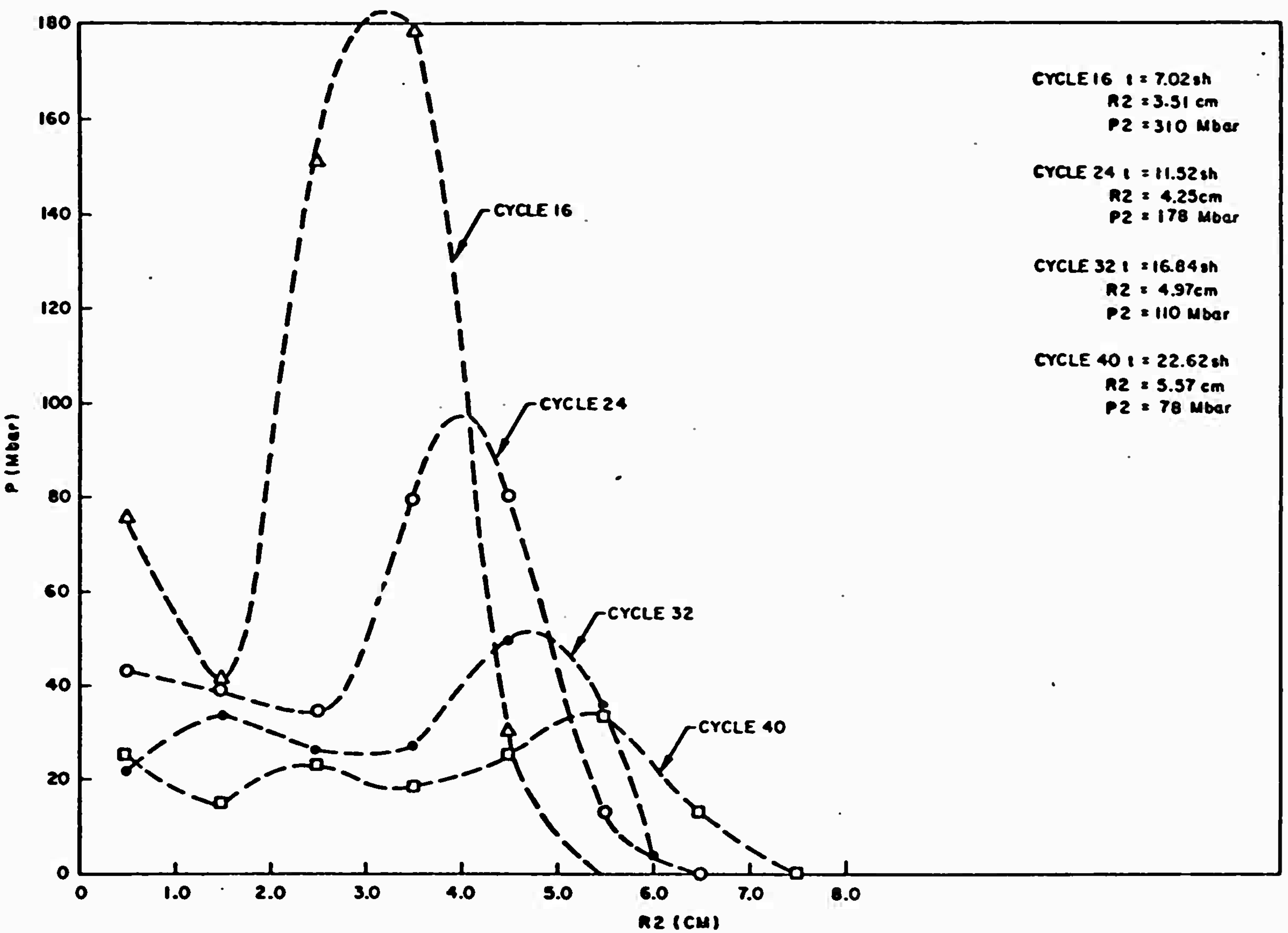


Fig. 4--Comparison of theory and computer results for a strong shock-point source solution

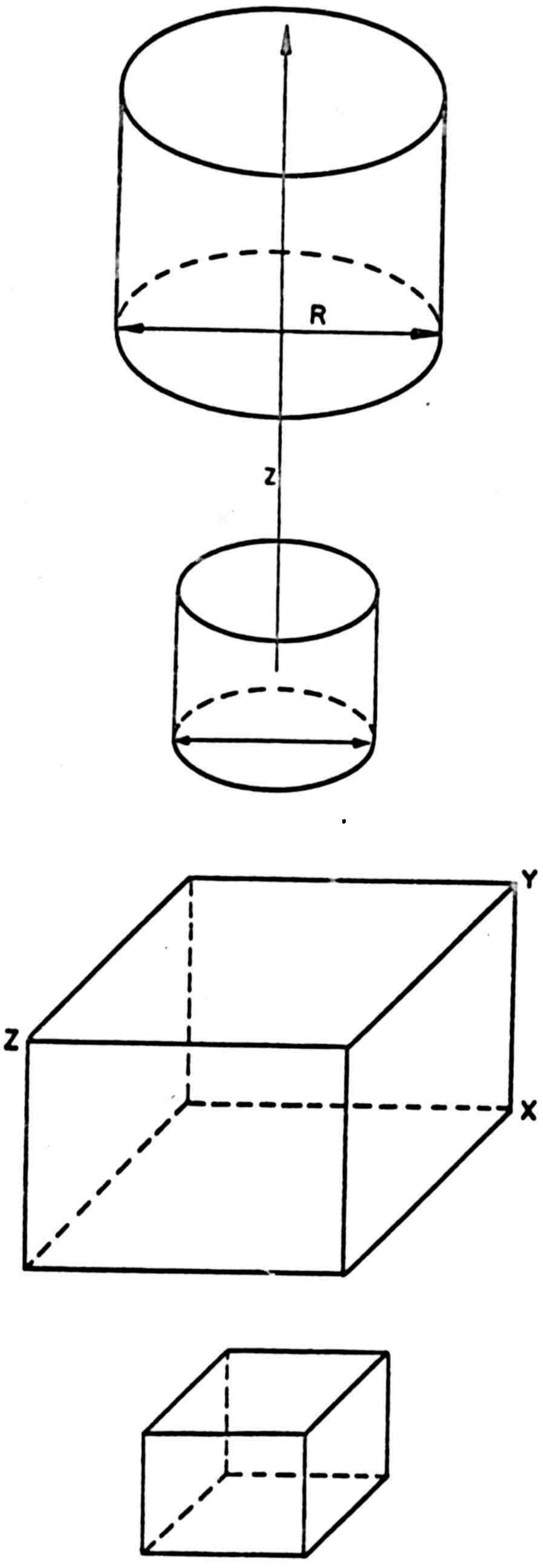


Fig. 5--Initial configuration of the OIL (top) and TRIOIL (bottom) problems

The configuration for OIL consisted of a right circular cylinder (D = L) where the diameter was equal to the side of the cube in TRIOIL. The OIL problem had 2 radial zones and 4 axial in the projectile. The velocity of the projectile was 2.6×10^6 cm/sec, and the projectile and target were both aluminum.

The momentum and times of the two problems were scaled by the ratio of their masses, the distances by the cube root of the ratio. Figure 6 presents the total positive y momentum (axial, in the case of OIL) versus time for the 2 codes. The agreement is excellent. Figure 7 displays the pressure attenuation into the target (along the axis for OIL and in either of the 2 inner original columns for the TRIOIL). Here again, we expect some difference at early times since one projectile is a cylinder and the other a cube, but the agreement is very good.

Figure 8 is a velocity (in the x-y plane) plot for the plane of symmetry, z = 0. Figure 9 is a velocity plot (in the y-z plane at a value of x in the center of the projectile. Exact symmetry is indicated by these two plots. Figure 10 is at a later time, presenting the velocity (in the x-y plane) plot for the plane of symmetry, z = 0.

Figure 11 is a velocity (in the x-y plane) plot for the plane of symmetry, z = 0. for an oblique impact (45°). The dark lines indicate the position of the original projectile to target at time t = 0. Figure 12 is a velocity plot in the y-z plane at a value of x at the right hand side of the original projectile.

Since the original formulation of the TRIOIL code, investigations into the effect of obliquity, and velocity, for semi-infinite and thin targets have been completed successfully.

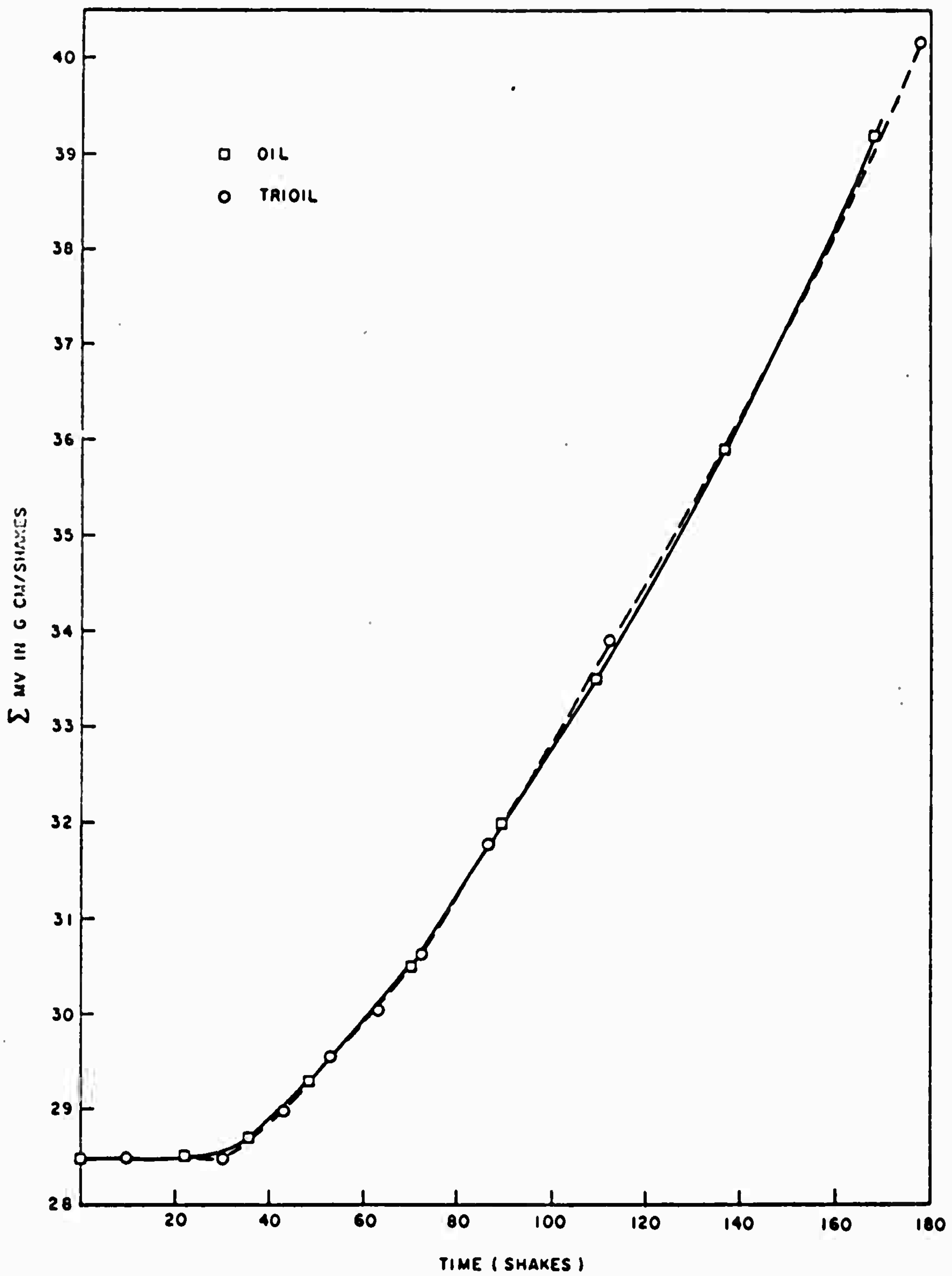


Fig. 6--Total positive Y momentum as a function of time for the two codes

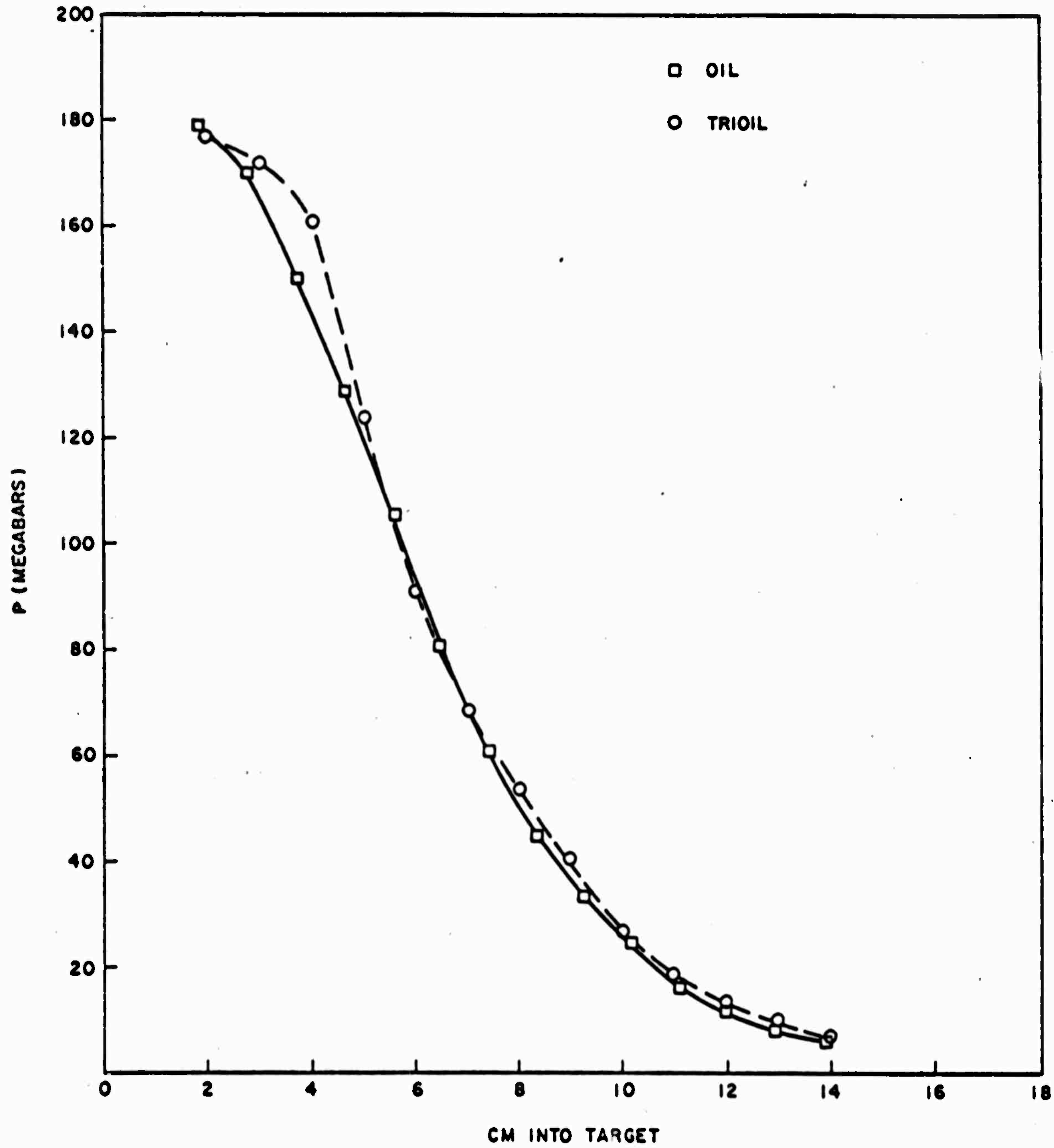


Fig. 7--Pressure attenuation into the target comparison for the 2 codes

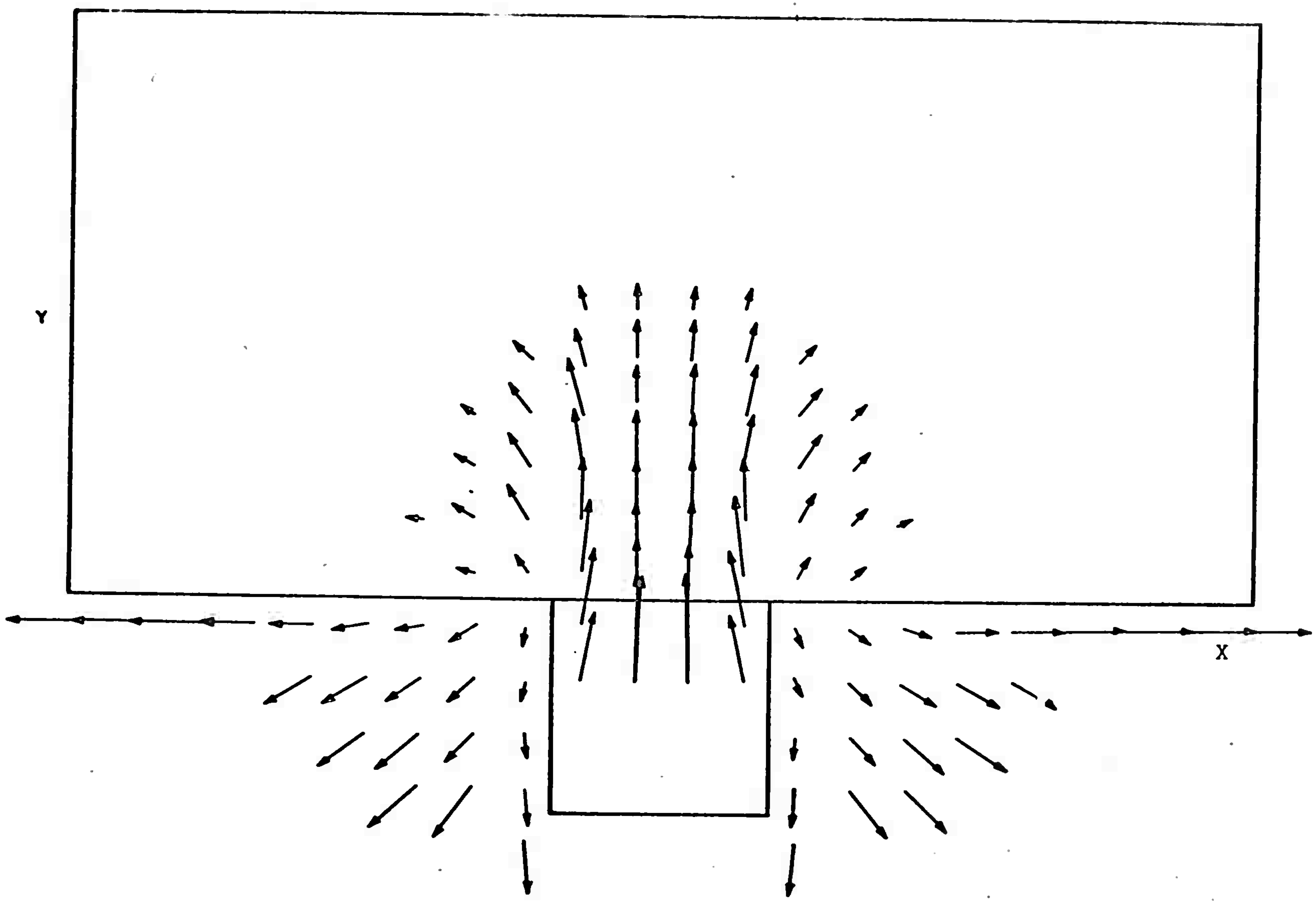


Fig. 8--Velocities (X-Y) in the plane of symmetry (Z=0) for 90 degrees

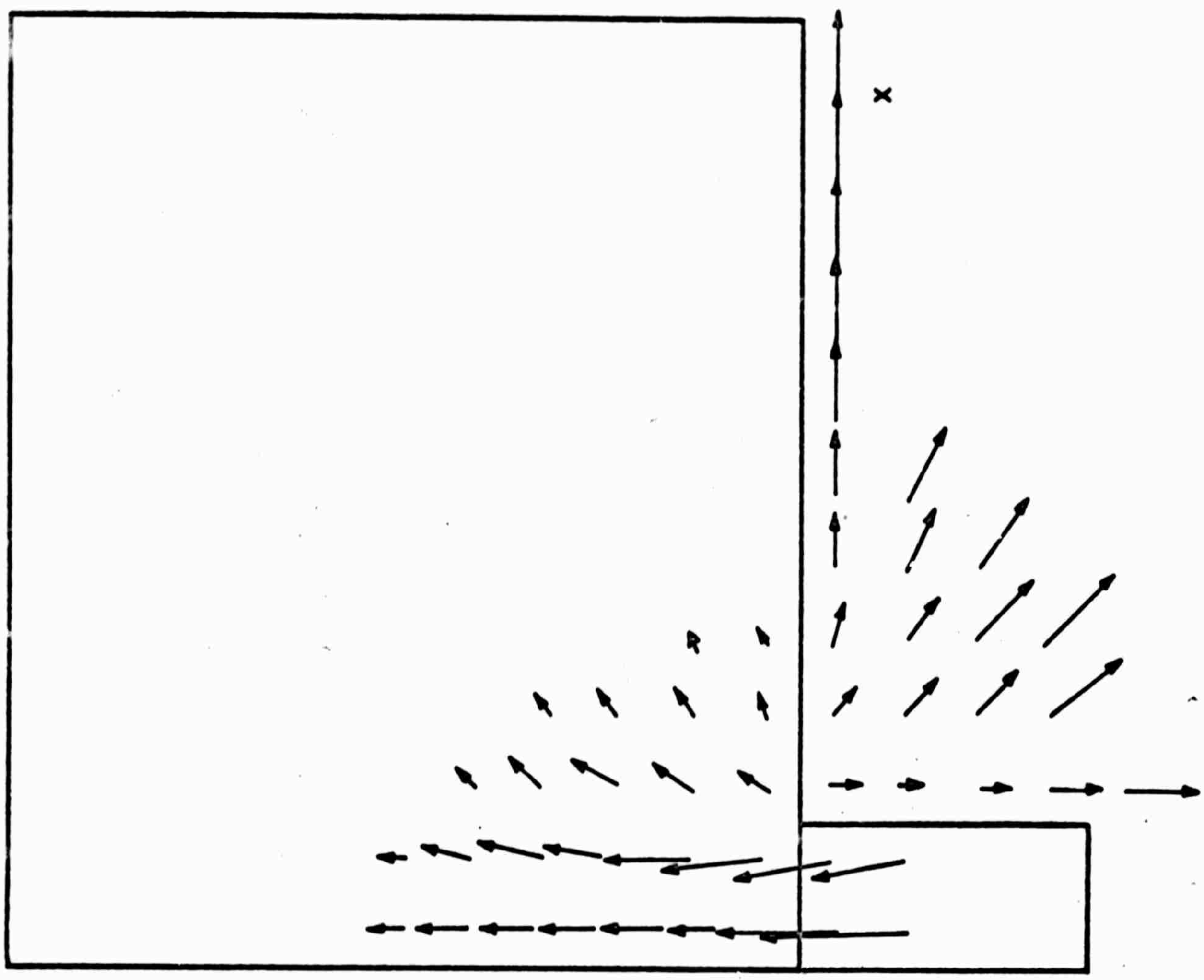


Fig. 9--Velocities (Y-Z) in the Y-Z plane at a X position of the original cente. of projectile for 20 degrees

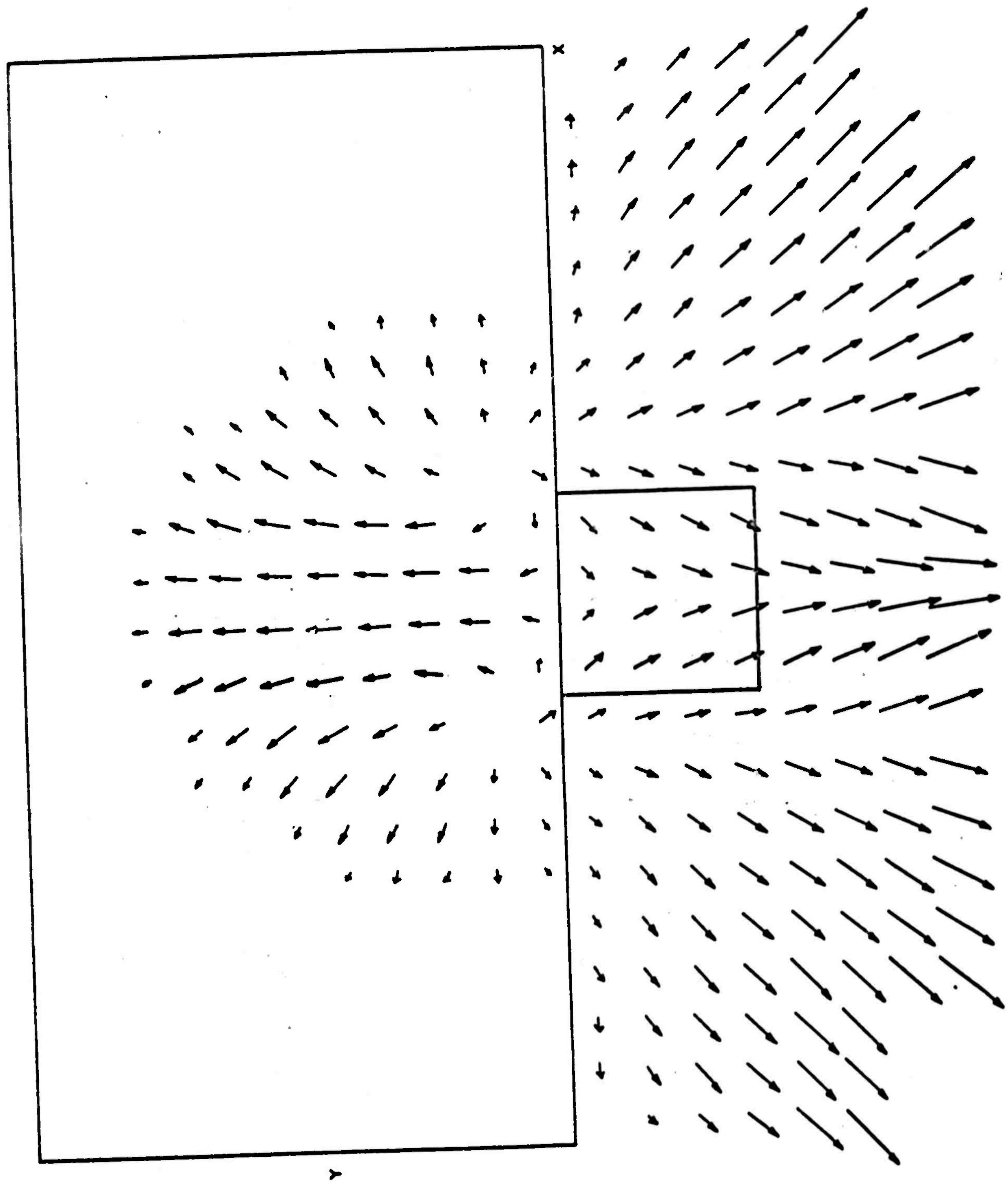


Fig. 10--Velocities (X-Y) in the plane
of symmetry (Z=0) for 90 degrees

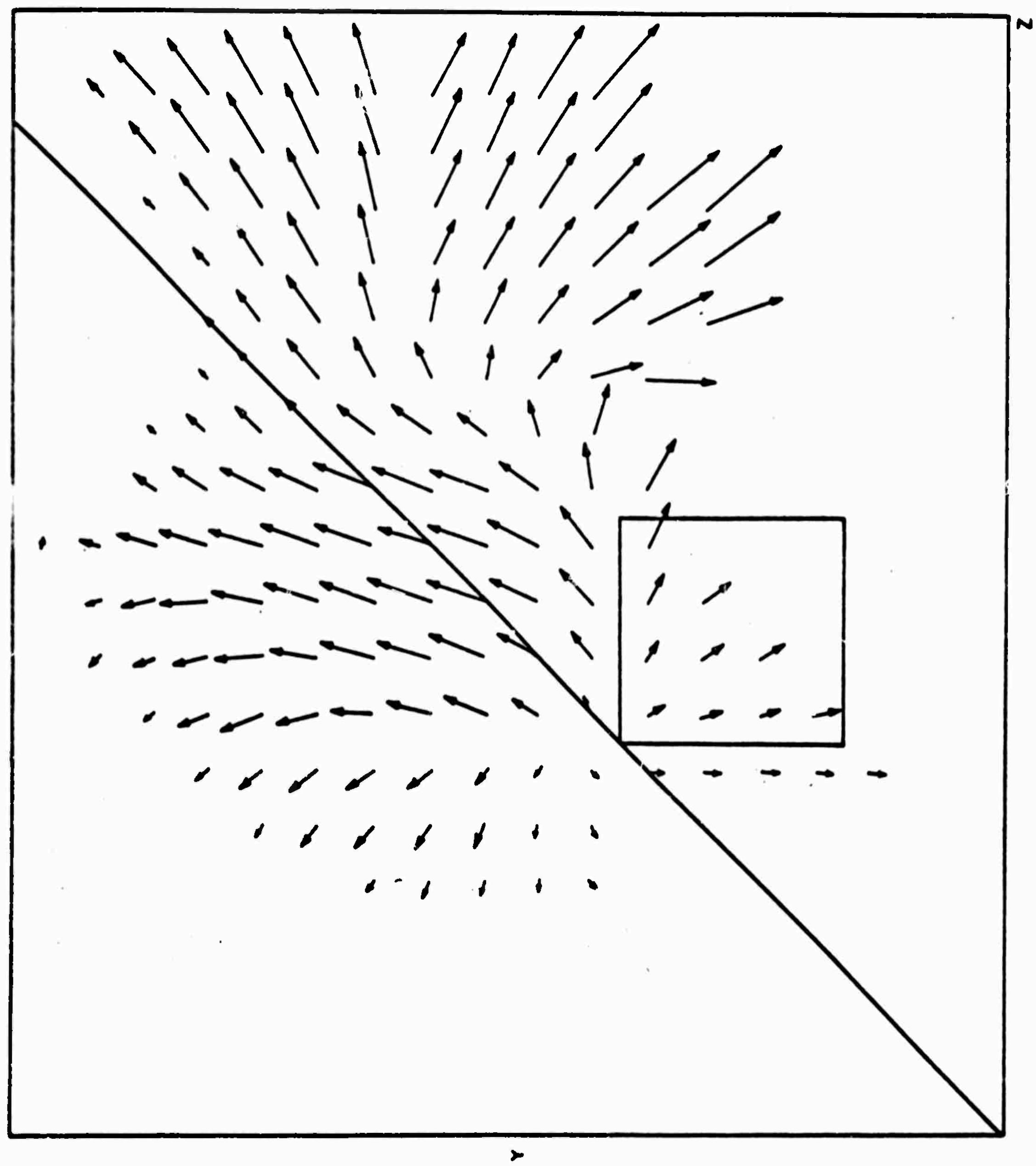


Fig. 11--Velocities (X-Y) in the plane of symmetry (Z=0) for 45 degrees

4. FORTRAN LISTING

```
      1   WALLY                                                    SV 07-25-67 1720
C
C
C      ***** NOTE, THE FOLLOWING SET OF DIMENSION,
C       COMMON, AND EQUIVALENCE IS TO BE USED FOR ALL
C       SUBROUTINES WITH THE EXCEPTION OF SUBROUTINE CARDS....................
C
       DIMENSION  AIX(6000),AMX(6000),U(6000),V(6000),W(6000),P(6000),
      1DX(30),DY(30),DZ(30),UL(30),PL(30),X(30),Y(30),ZCOR(30),
      2Z(150),IZ(150),FLEFT(30),YAMC(30),SIGC(30),GAMC(30),ZMOM(30),
      3PBIND(700),UBIND(700),BMASS(700),BXMOM(700),BYMOM(700),
      4BZMOM(700),BENR(700)
       DIMENSION PR(50),PK(30)
       COMMON  Z,AIX,AMX,U,V,W,P,DX,DY,DZ,
      1UL,PL,X,Y,ZCOR,PR,SIGC,GAMC,
      2ZMOM,UBIND,BMASS,BYMOM,
      3BZMOM,BENR,AREA,BIG,BOUNCE,PABOVE,PBLO,
      4PIDTS,PRR,RHO,SIG,UVMAX,VABOVE,VBLO,
      5VEL,WPS,WS,WSA,WSB,WSC,I,II,IN,IR,
      6IWS,IWSA,IWSB,IWSC,J,JN,JP,JR,K,KDT,KN,
      7KP,KR,KRM,L,M,MA,MB,MC,MD,ME,MZ,N
       COMMON REZ,TRAD,DTRAD,RADEB,RADER,RADET,X1,X2,Y1,Y2,IMAXA
       EQUIVALENCE (Z,IZ,PROB),(Z(2),CYCLE),(Z(3),DT),
      1(Z(4),PRINTS),(Z(5),PRINTL),(Z(6),DUMPT7),(Z(7),CSTOP),(Z(8),PIDY)
      2,(Z(9),GAM),(Z(10),GAMD),(Z(11),GAMX),(Z(12),ETH),(Z(13),FFA),
      3(Z(14),FFB),(Z(15),TMASS),(Z(16),XMAX),(Z(17),YMAX),(Z(18),ZMAX),
      4(Z(19),DNN),(Z(20),DMIN),(Z(21),DTNA),(Z(22),REZFCT),(Z(23),TOZONE
      5),(Z(24),ECK),(Z(25),SBOUND),(Z(26),CABLN),(Z(27),T),(Z(28),GMAX),
      6(Z(29),WSGD),(Z(30),WSGX),(Z(31),GMADR),(Z(32),GMAXR),(Z(45),DTCHK
      7),(Z(46),PCSTAB),(Z(47),CNOT),(Z(48),BFACT),(Z(49),EPSI),(Z(50),S1
      8),(Z(51),S2),(Z(52),S3),(Z(53),S4),(Z(54),S5),(Z(55 ,S6),
      9(Z(56),S7),(Z(57),S8),(Z(58),S9),(Z(59),S10)
       EQUIVALENCE (Z(60),AMLOST),(Z(61),ELOST),(Z(62),XMLOST),
      1(Z(63),YMLOST),(Z(64),ZMLOST),(Z(65),ENEG),(Z(66),RHONOT),
      2(Z(67),VELOC),(Z(68),BUG),(Z(81),NPR),(Z(82),NPRI),
      3(Z(83),NC),(Z(84),NPC),(Z(85),NRC),(Z(86),IMAX),(Z(87),JMAX),
      4(Z(88),KMAX),(Z(89),KMAXA),(Z(90),IXMAX),(Z(91),NOD),
      5(Z(92),NOPR),(Z(93),I1),(Z(94),I2),(Z(95),I3),(Z(96),I4),
      6(Z(97),N1),(Z(98),N2),(Z(99),N3),(Z(100),N4),(Z(101),N5),
      7(Z(102),N6),(Z(103),N7),(Z(104),N8),(Z(105),N9),(Z(106),N10),
      8(Z(107),N11),(Z(108),K1),(Z(109),K2),
      9(Z(110),J1),(Z(111),J2)
       EQUIVALENCE (BMASS,PBIND),(BXMOM,UBIND),
      1(UL,FLEFT),(PL,YAMC,PK)
C
C
C
@I FOR   MAIN/S,MAIN/S,MAIN/SS
C                                                                       INPU0760
C
C      ********* 3DOIL **************************
C
C                                                                       MAIN0050
C       THE INPUT ROUTINE WILL READ A TRIOIL  BINARY DUMP
C        TAPE OR WILL CALL FOR SUBROUTINE SET-UP
C           WHICH WILL GENERATE A DATA TAPE
C              FOR RESTRICTED GEOMETRY
       CALL INPUT                                                       MAIN0060
```

```
C
C        CDT ROUTINE CALLS FOR THE EQUATION OF STATE,
C           CALCULATES THE DT(HYDRODYNAMICS TIME STEP) AND
C             ADVANCES THE CYCLE NUMBER
   10 CALL CDT                                                          MAIN0070
C
C          IN EDIT , DETERMINE WHETHER TO EXECUTE A
C            SHORT PRINT, LONG PRINT, BINARY TAPE DUMP
C                 OR TO STOP THE PROBLEM
      CALL EDIT                                                         MAIN0080
C         *** SENSE LITE 1 SIGNIFIES THAT THIS
C             IS THE LAST CYCLE OF THIS RUN.
C          THE LITE IS TURNED ON IN THE EDIT ROUTINE...........
      CALL SLITET(1,K000FX)                                             MAIN0090
       GO TO(30,20),K000FX                                              MAIN0100
C
C          PH1 INTEGRATES THE MOMENTUM AND
C             ENERGY EQUATIONS DUE TO
C             TO PRESSURE FORCES ONLY , THE CONVECTIVE
C            TERMS ARE TEMPORARILY DROPPED.
   20 CALL PH1                                                          MAIN0110
C
C           PH2 SOLVES THE MASS TRANSPORT EQUATIONS
C           AND MOVES MOMENTUM AND ENERGY
C            ACROSS THE FIXED CELL BOUNDARIES
C         TO APPROXIMATE THE CONVECTIVE TERMS THAT
C              WERE OMITTED IN PH1
      CALL PH2                                                          MAIN0120
C                                                                       MAIN0130
C         ALL CELL QUANTITIES , EXCEPT THE PRESSURE
C              HAVE NOW BEEN ADVANCED TO
C         CYCLE N+1
C                                                                       MAIN0140
      GO TO 10                                                          MAIN0150
   30 CALL EXIT
      END                                                               MAIN0170
@I FOR   CARDS/S,CARDS/S,CARDS/SS
      SUBROUTINE CARDS                                                  CARD0010
      DIMENSION TABLE(1),CARD(7),LABLE(1)                               CARD0020
      COMMON         TABLE                                              CARD0030
C     A 2 IN COLUMN 1, ROUTINE WILL FIX THE
C     FLOATING PT. NO.
C     A 1 IN COLUMN 1, MEANS THIS IS LAST CARD TO
C     READ IN.
      EQUIVALENCE(TABLE(1),LABLE(1))                                    CARD0050
      WRITE(6,10)
    1 READ  (5,11)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)                  CARD0080
      WRITE(6,12)IEND,LOC,NUMWPC,(CARD(I),I=1,NUMWPC)
      DO 4 I=1,NUMWPC                                                   CARD0100
      J=LOC+I-1                                                         CARD0110
      IF(IEND-2)2,5,2                                                   CARD0120
    5 LABLE(J)=IFIX(CARD(I))                                            CARD0130
      GO TO 4                                                           CARD0140
    2 TABLE(J)=CARD(I)                                                  CARD0150
    4 CONTINUE                                                          CARD0160
      IF(IEND-1)1,3,1                                                   CARD0170
    3 RETURN                                                            CARD0180
```

```
C                    FORMATS                                         CARD0190
   10 FORMAT(20H1 TRIOIL INPUT CARDS///)
   11 FORMAT(I1,I5,I1,0P7E9.4)                                       CARD0210
   12 FORMAT(1H I4,I7,I3,1P7E14.6)                                   CARD0220
      END                                                            CARD0230
@I FOR   UNCLE/S,UNCLE/S,UNCLE/SS
      SUBROUTINE UNCLE
      REWIND N7
C     SUBROUTINE (( UNCLE))) IS CALLED WHENEVER
C     A CODED ERROR IS ENCOUNTERED
C      IN ANY SUBROUTINE, ITS MAIN FUNCTION
C       IS TO PRINT THE CELL QUANTITES OUT IN THE FORM
C      OF THE NORMAL LONG PRINT
      NR=90
      WRITE(6,8120)NC
 8120 FORMAT(1H0//68H AN ERROR HAS OCCURRED AND SUBROUTINE UNCLE HAS BEE
     1N CALLED AT CYCLEI5////)
 5000 WRITE(6,8116)PROB,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
      DO 1126 KK=K1,K2
      WRITE(6,9041)KK,ZCOR(KK),DZ(KK)
 5004 DO 5050 I=I1,I2
      WS1=1.
      J=J2+1
      K=J2*IMAX+I+(KK-1)*IXMAX
      DO 5046 L=J1,J2
      J=J-1
      K=K-IMAX
 5012 IF(AMX(K))5046,5046,5014
 5014 IF(WS1)5019,5019,5016
 5016 WRITE(6,8135)I,X(I),DX(I)
      WS1=0.
 5019 WS=AMX(K)/(DX(I)*DY(J)*DZ(KK))
      WSC=P(K)*1.E+4
 5018 WRITE(6,8108)J,U(K),V(K),WSC,AMX(K),WS,AIX(K),W(K),Y(J)
 5046 CONTINUE
 5050 CONTINUE
 1126 CONTINUE
      RETURN
 8108 FORMAT(I3,1X,1P8E12.5)
 81160FORMAT(8H1PROBLEM6X,5HCYCLE9X,4HTIME13X,2HDT13X,4HTRAD11X,5HDTRAD1EDIT3380
     12X,2HNR6X,2HN14X,2HN24X,2HN34X,2HN4/(F7.1,I11,2X,1P4E16.7,I10,2X,4EDIT3390
     2I6))
 81350FORMAT(1H ///4H I =I3,6X,6HX(I) =F12.3,6X,7HDX(I) =F12.3//3H  J8X,EDIT3520
     11HX10X,1HY10X,3HF/A9X,3HAMX9X,3HRHO8X,3HAIX9X,4H   W 8X,2H Y/)
 9041 FORMAT(1H ///4H K =I3,6X,9HZCOR(K) =F12.3,6X,7HDZ(K) =F12.3)
      END
@I FOR   SETUP/S,SETUP/S,SETUP/SS
      SUBROUTINE SETUP
C
C     CALCULATE THE ADDITIONAL INDICES THAT ARE FUNCTIONS OF
C         IMAX,JMAX, AND KMAX
C
      IXMAX=(IMAX)*(JMAX)
      KMAXA= KMAX*IXMAX
C
C     SET ALL CELL CENTERED QUANTITIES TO ZERO
C
```

```
      DO 1 K=1,KMAXA
      U(K)=0.
      V(K)=0.
      W(K)=0.
      AIX(K)=0.
      AMX(K)=0.
      P(K)=0.
    1 CONTINUE
      X(1)=DX(1)
C
C      CALCULATE ALL X,S
      DO 10 I=2,IMAX
      X(I)=X(I-1)+DX(1)
C
C     NOTE, DX IS CONSTANT FOR ALL I
      DX(I)=DX(1)
   10 CONTINUE
      Y(1)=DY(1)
C
C     CALCULATE ALL Y,S
      DO 11 J=2,JMAX
      Y(J)=Y(J-1)+DY(1)
C
C     NOTE, DY IS CONSTANT FOR ALL J
      DY(J)=DY(1)
   11 CONTINUE
      ZCOR(1)=DZ(1)
C
C     CALCULATE ALL ZCOR,S
      DO 12 K=2,KMAX
      ZCOR(K)=ZCOR(K-1)+DZ(1)
C
C      NOTE, DZ IS CONSTANT FOR ALL K
      DZ(K)=DZ(1)
   12 CONTINUE
C
C      RHONOT IS INITIAL DESITY FOR ALL MATERIAL
      J3=S1
      DO 100 K=1,KMAX
      LL=(K-1)*IXMAX
      DO 100 I=1,IMAX
      DO 100 J=J3,JMAX
      L=LL+(J-1)*IMAX+I
      AMX(L)=DX(I)*DY(J)*DZ(K)*RHONOT
  100 CONTINUE
C
C     S1=INTERFACE(J) VALUE +1 BETWEEN PROJECTILE AND TARGET
C     S2=BACK BOUNDARY +1 OF THE PROJECTILE(K) VALUE
C     S3= FRONT BOUNDARY OF PROJECTILE(K) VALUE
C     S4= LEFT BOUNDARY(I) VALUE +1 OF THE PROJECTILE
C     S5= RIGHT BOUNDARY(I) OF THE PROJECTILE
C     S6= BOTTOM BOUNDARY (J) +1 OF THE PROJECTILE
C     S7= TOP BOUNDARY(J) OF THE PROJECTILE
      ETH=0.
      K11=S2
      K22=S3
      I11=S4
```

```
      I22=S5
      J11=S6
      J22=S7
      DO 200 K=K11,K22
      LL=(K-1)*IXMAX
      DO 200 I=I11,I22
      DO 200 J=J11,J22
      L=LL+(J-1)*IMAX+I
      AMX(L)=DX(I)*DY(J)*DZ(K)*RHONOT
C     VELOC=INITIAL Y COMPONENT OF VELOCITY
      V(L)=VELOC
C     S9= INITIAL X COMPONENT OF VELOCITY
      U(L)=S9
C     S10= INITIAL Z COMPONENT OF VELOCITY
      W(L)=S10
      ETH=ETH+AMX(L)*(U(L)**2+V(L)**2+W(L)**2)/2.
  200 CONTINUE
C
C     PRINT THE QUANTITIES ASSOCIATED WITH THE GRID
C
      WRITE(6,8000)IMAX,(X(I),I=1,IMAX)
      WRITE(6,8003)IMAX,(DX(I),I=1,IMAX)
      WRITE(6,8001)JMAX,(Y(J),J=1,JMAX)
      WRITE(6,8004)JMAX,(DY(J),J=1,JMAX)
      WRITE(6,8002)KMAX,(ZCOR(K),K=1,KMAX)
      WRITE(6,8005)KMAX,(DZ(K),K=1,KMAX)
      WRITE(6,8006)IMAX,JMAX,KMAX,IXMAX,KMAXA
 8000 FORMAT(1H /10H X(I)  I=1,I2/(5F16.6))
 8001 FORMAT(1H /10H Y(J)  J=1,I2/(5F16.6))
 8002 FORMAT(1H /13H ZCOR(K)  K=1,I2/(5F16.6))
 8003 FORMAT(1H /11H DX(I)  I=1,I2/(5F16.6))
 8004 FORMAT(1H /11H DY(J)  J=1,I2/(5F16.6))
 8005 FORMAT(1H /11H DZ(K)  K=1,I2/(5F16.6))
 8006 FORMAT(7I8)
C
C     WRITE A DUMP TAPE(FOR T=0.)  FOR
C      THE TRIOIL CODE
      REWIND N7
      WS=555.0
      WRITE(N7)WS,CYCLE,N3
      WRITE(N7)(Z(I),I=1,150)
      WRITE(N7)(U(I),V(I),W(I),AMX(I),AIX(I),I=1,KMAXA)
      WRITE(N7)(X(I),I=1,IMAX)
      WRITE(N7)(Y(J),J=1,JMAX)
      WRITE(N7)(ZCOR(K),K=1,KMAX)
      WS=666.
      WRITE(N7)WS,WS,WS
      RETURN
      END
@I FOR   INPUT/S,INPUT/S,INPUT/SS
      SUBROUTINE INPUT                                                  INPU0010
C                                                                       MAIN0020
C                                                                       INPU0900
      CALL SLITE (3)                                                    INPU0980
C      IWSA=THE NUMBER OF BCD HEADER CARDS TO
C                             READ IN
      READ(5,8007)IWSA
```

```
      DO 7 I=1,IWSA
      READ  (5,8004)IWS                                               INPU1000
      WRITE(6,8004)IWS
    7 CONTINUE
    6 CALL CARDS                                                      INPU1020
C      NOTE, PROVSIONS FOR CALLING A GENERATOR
C         CODE SUCH AS (CLAM) , DOES  NOT EXIST IN
C             3 DIMENSIONS AT THIS TIME
      IF(PK(3))8887,8888,8888
 8888 CALL CARDS
      CALL SETUP
 8887 CONTINUE                                                        INPU1060
      GO TO 1000                                                      INPU1090
   10 CONTINUE                                                        INPU1120
      CALL CARDS                                                      INPU1130
      GO TO 2000                                                      INPU1140
   40 DO 45 K=1,KMAXA                                                 INPU1160
   45 P(K)=0.0                                                        INPU1170
C
C     INTEGRATE(THE  TIME AND CYCLE NUMBER ) BACKWARDS, SINCE THEY
C     WILL BE ADVANCED IN  CDT SUBROUTINE
C
      T=T-DTNA                                                        INPU1180
      NC=NC-1                                                         INPU1190
      CYCLE=NC                                                        INPU1200
      NPC=NPC-1                                                       INPU1210
      UVMAX=0.0                                                       INPU1220
C
C     CALCULATE DX,DY,DZ SINCE THEY ARE NOT ON THE TAPE
C
C
      WS=0.
      DO 50 I=1,IMAX                                                  INPU1230
      DX(I)=X(I)-WS
      WS=X(I)
   50 CONTINUE
      WS=0.
      DO 55 J=1,JMAX                                                  INPU1250
      DY(J)=Y(J)-WS
      WS=Y(J)
   55 CONTINUE
      WS=0.
      DO 65 K=1,KMAX
      DZ(K)=ZCOR(K)-WS
      WS=ZCOR(K)
   65 CONTINUE
C     DUMP THE  Z BLOCK
      K=1
      DO 80 I=1,9
      L=K+8
      WRITE(6,8005)K,(Z(N),N=K,L)
   80 K=L+1
      K=81
      DO 81 I=1,8
      L=K+8
      WRITE(6,8006)K,(IZ(N),N=K,L)
   81 K=L+1
```

```
      GO TO 10000                                                       INPU1370
C                                                                       INPU1380
C                                                                       INPU1390
C     READ THE BINARY TAPE FOR A RESTART
C                                                                       INPU1400
 1000 MZ=150                                                            INPU1410
      IWS=0                                                             INPU1420
 1003 REWIND N7
 1004 READ(N7)PR(1),PR(2),N3
      NR=N3+6
 1006 IF(PR(1)-555.0)1010,1016,1010                                     INPU1460
 1010 IWS=IWS+1                                                         INPU1470
 1011 IF(MOD(IWS,3))9902,9902,1003                                      INPU1480
 1016 IF(PR(2))1010,1018,1018                                           INPU1490
 1018 IF(PR(2)-PR(2))1023,1023,1020                                     INPU1500
C     READ OVER ,THIS IS NOT THE CORRECT CYCLE
 1020 DO 1022 L=2,NR                                                    INPU1510
 1022 READ(N7)WSS
      GO TO 1004                                                        INPU1530
 1023 READ(N7)(Z(I),I=1,MZ)
C     CHECK FOR CORRECT PROBLEM NUMBER
      IF(ABS(PROB-PR(1))-.01)1024,1024,9901                             INPU1550
 1024 READ(N7)(U(I),V(I),W(I),AMX(I),AIX(I),I=1,KMAXA)
      READ(N7)(X(I),I=1,IMAX)
      READ(N7)(Y(J),J=1,JMAX)
      READ(N7)(ZCOR(K),K=1,KMAX)
 1034 READ(N7)PR(1),PR(2),PR(3)
 1036 IF(PR(1)-555.0)9904,1040,1038                                     INPU1680
 1038 IF(PR(2)-666.0)9905,1040,9905                                     INPU1690
 1040 GO TO 10                                                          INPU1700
 2000 IF(WSGX)9906,2010,2005                                            INPU1750
C     CALCULATE 1./(GAMMA-1.) AND  GAMMA/(GAMMA-1.)
 2005 GAMX=1.0/(WSGX-1.0)                                               INPU1760
 2010 WSGX=(GAMX+1.0)/GAMX                                              INPU1770
      GMAXR=GAMX*WSGX                                                   INPU1780
 2012 IF(WSGD)9907,2020,2015                                            INPU1790
 2015 GAMD=1.0/(WSGD-1.0)                                               INPU1800
 2020 WSGD=(GAMD+1.0)/GAMD                                              INPU1810
      GMADR=GAMD*WSGD                                                   INPU1820
      GMAX=WSGD                                                         INPU1830
C     SEARCH FOR MAX GAMMA
      IF(WSGD-WSGX)2025,2030,2030                                       INPU1840
 2025 GMAX=WSGX                                                         INPU1850
 2030 GO TO 40                                                          INPU1860
C               ERROR                                                   INPU1900
 9901 NK=1023                                                           INPU1910
      GO TO 9999                                                        INPU1920
 9902 NK=1011                                                           INPU1930
      GO TO 9999                                                        INPU1940
 9904 NK=1036                                                           INPU1950
      GO TO 9999                                                        INPU1960
 9905 NK=1038                                                           INPU1970
      GO TO 9999                                                        INPU1980
 9906 NK=2000                                                           INPU1990
      GO TO 9999                                                        INPU2000
 9907 NK=2012                                                           INPU2010
 9999 NR=1                                                              INPU2020
```

```
      WRITE(6,8000)PR(1),PR(2),PK(1),PK(2),PK(3),IWS
      CALL UNCLE
      CALL DUMP
C.                                                                      INPU2040
10000 RETURN                                                            INPU2050
C                                                                       INPU2060
C                  FORMATS                                              INPU2070
 8000 FORMAT(1P5E14.6,I5)
 80040FORMAT(I1,71H                                                     INPU2090
     1                                       )                          INPU2100
 8005 FORMAT(I4,1X,1P9E12.5)
 8006 FORMAT(I4,1X,9I7)
 8007 FORMAT(6I3)
C                                                                       INPU2120
      END                                                               INPU2130
@I FOR   CDT/S,CDT/S,CDT/SS
      SUBROUTINE CDT                                                    CDT 0010
C                                                                       CDT 0020
 3000 VEL=0.0                                                           CDT 1030
C
C     SET UP THE LOOPS  FOR CALCULATING THE  MATERIAL PRESSURE
C
      DO 3050 K=K1,K2
      LL=(K-1)*IXMAX
 3005 DO 3050 I=I1,I2
 3010 L=I+LL+(J1-1)*IMAX
 3015 DO 3050 J=J1,J2
 3020 IF(AMX(L))9901,3050,3025
C
C     THE ES ROUTINE CALCULATES THE PRESSURE, FOR INPUT IT NEEDS I,J,K,L
C     AND THE  AIX(ENERGY) AND AMX(MASS)
C
 3025 CALL ES                                                           CDT 1110
C
C     THE OUTPUT FROM ES  IS  THE   P(PRESSURE) AND  MAX (GAMMA-1.)
C
 3030 IF(ABS(P(L))-1.E-20)3035,3035,3040
 3035 P(L)=0.
 3040 IF(WSGX-VEL)3050,3050,3045                                        CDT 1150
 3045 VEL=WSGX                                                          CDT 1160
 3050 L=L+IMAX
 3055 KDT=1                                                             CDT 1180
      UVMAX=-1.0                                                        CDT 1190
C
C     NOW SET UP THE LOOP FOR CALCULATING DT FROM THE PARTICLE
C     VELOCITIES AND THE COURANT CONDITION
C
C     WE FLAG THE CELL THAT IS CONTROLLING THE TIME STEP
C        AND STORE THE VALUES OF I,J,K INTO
C        N10,N11, AND N9 RESPECTIVELY.
C
      DO 3255 K=K1,K2
      LL=(K-1)*IXMAX
 3070 DO 3255 I=I1,I2
 3075 L=I+LL+(J1-1)*IMAX
 3095 DO 3255 J=J1,J2
 3100 CONTINUE
```

```
      IF(AMX(L))9901,3255,4
C     IF RHO IS LESS THAN DTCHK, OMIT THE STABILITY CHECK FOR THIS CELL
    4 IF(AMX(L)/(DX(I)*DY(J)*DZ(K))-DTCHK)3255,3255,3115
 3115 SIG=DX(I)                                                         CDT 1260
 3120 IF (DY(J)-SIG)3125,3130,3130                                      CDT 1270
 3125 SIG=DY(J)                                                         CDT 1280
 3130 IF(DZ(K)-SIG)5131,5130,5130
 5131 SIG=DZ(K)
C     HERE WE CALCULATE THE SPEED OF SOUND , THE PERFECT GAS SPEED OF
C     SOUND OR  DP/DRHO FROM THE METAL EQUATION OF STATE
 5130 IF(CNOT)4000,4000,4001
 4000 WS=SQRT(GMAX*P(L)/AMX(L)*DX(I)*DY(J)*DZ(K))
      GO TO 3205                                                        CDT 1310
 4001 WSA=ABS(P(L))*1.E+4
      WS=CNOT+SFACT*(WSA**EPSI)
      WS=WS*1.E-3                                                       CDT 1340
 3205 WS=WS/SIG                                                         CDT 1350
 3210 IF(UVMAX-WS)3215,3220,3220                                        CDT 1360
 3215 N10=I                                                             CDT 1370
      N11=J                                                             CDT 1380
      N9=K
      UVMAX=WS                                                          CDT 1390
 3220 CONTINUE
    2 WS=ABS(U(L))/DX(I)
 3225 IF(UVMAX-WS)3230,3235,3235                                        CDT 1450
 3230 UVMAX=WS                                                          CDT 1460
      N10=I                                                             CDT 1470
      N9=K
      N11=J                                                             CDT 1480
 3235 WS=ABS(V(L))/DY(J)
 3240 IF(UVMAX-WS)3245,3250,3250                                        CDT 1500
 3245 N10=I                                                             CDT 1510
      N11=J                                                             CDT 1520
      N9=K
      UVMAX=WS                                                          CDT 1530
 3250 CONTINUE                                                          CDT 1540
      WS=ABS(W(L))/DZ(K)
      IF(UVMAX-WS)5245,5250,5250
 5245 N9=K
      N10=I
      N11=J
      UVMAX=WS
 5250 CONTINUE
 3255 L=L+IMAX
      IF(UVMAX)9912,9912,3260
C
C     HERE,CHECK THE  3 POSSIBLE OPTIONS FOR CALCULATING THE
C                    NEW  DT
C
 3260 IF(CALLN)90,91,3300                                               CDT 1560
C          HERE OPTIONS EXIST FOR STARTING THE PROBLEM
C          WITH SMALL TIME STEPS(A SMALL FRACTION OF STABILITY)
C          FOR PROBLEMS WHERE THE INITIAL ENERGY IS
C           PRIMARILY INTERNAL.............
   90 IF(Z(69)-1.0)93,94,94
   93 Z(69)=Z(69)*Z(70)
      GO TO 95
```

```
   94 Z(69)=1.0
   95 DT=.5/VEL/UVMAX*PCSTAB*Z(69)
      GO TO 3295                                                        CDT 1580
   91 WS=UVMAX*DT                                                       CDT 1590
      WSA=0.5/VEL                                                       CDT 1600
 3265 IF(FFA-WSA)3276,3276,3270                                         CDT 1610
 3270 FFA=WSA                                                           CDT 1620
 3276 IF(WS-FFA)3285,3300,3280                                          CDT 1630
 3280 DT=DT/WS*FFB/0.9                                                  CDT 1640
      GO TO 3295                                                        CDT 1650
 3285 IF(WS-FFB)3290,3290,3300                                          CDT 1660
 3290 DT=DT*FFA/WS*0.9                                                  CDT 1670
 3295 KDT=0                                                             CDT 1680
 3300 T=T+DTNA                                                          CDT 1690
   82 NC=NC+1                                                           CDT 1780
      CYCLE=NC                                                          CDT 1790
      UVMAX=UVMAX*DT
      NPC=NPC+1                                                         CDT 1800
 3305 IF(T)9909,3320,3310                                               CDT 1810
 3310 IF(KDT)9910,3315,3320                                             CDT 1820
 3315 WRITE(6,8000)T,DTNA,DT
 3320 DTNA=DT                                                           CDT 1840
      GO TO 3325                                                        CDT 1850
 9901 NK=3020                                                           CDT 1870
      GO TO 9999                                                        CDT 1880
 9909 NK=3305                                                           CDT 1890
      GO TO 9999                                                        CDT 1900
 9910 NK=3310                                                           CDT 1910
      GO TO 9999                                                        CDT 1920
 9912 NK=1
      GO TO 9999
 9999 NR=2                                                              CDT 1940
      WRITE(6,8002)I,J,K,L,NK,NR,NC
      WRITE(6,8001)UVMAX,AMX(L),P(L),DT,VEL
 8001 FORMAT(1P5E14.6)
 8002 FORMAT(8I5)
      CALL UNCLE
      CALL DUMP
 3325 RETURN                                                            CDT 1960
 8000 FORMAT (17H0CHANGE DT ... T=1PE10.3,10H    DT(N)=1PE10.3,12H    DT
     1(N+1)=1PE10.3)
      END                                                               CDT 1990
@I FOR   PH1/S,PH1/S,PH1/SS
      SUBROUTINE PH1                                                    PH1 0010
C
C     THE VELOCITIES,ENERGIES,PRESSURE AND MASS ARE AT
C     CELL CENTERS
C     FIRST PASS ,CALCULATE U,V AND W FOR CYCLE N+1 ,AND THE  WORK
C     TERMS USING THE VELOCITIES AT CYCLE (N).
C     SECOND PASS, CALCULATE THE CONTRIBUTION TO THE CHANGE IN
C     INTERNAL ENERGY FROM WORK TERMS EVALUATED FROM  U,V AND W
C     AT CYCLE N+1
C     2 PASSES REQUIRED
C
C     REMEMBER, WE  ARE NOT ADVANCING THE VELOCITIES AND ENERGY TO CYCLE
C     N+1,SINCE  WE  HAVE NOT AS YET  SOLVED THE TRANSPORT TERMS, AS
C     USUAL, WE REFER TO THE VELOCITIES AND ENERGY FROM THIS ROUTINE
```

```
C     AS TILDA QUANTITIES
C
C     CELL IN QUESTION = L=(J-1)*IMAX+I+(K-1)*IMAX*JMAX
C
C     IXMAX=IMAX*JMAX
C
C     KMAXA=(IMAX)(JMAX)(KMAX)
C
C     CELL TO THE RIGHT= L+1
C
C     CELL ABOVE =N= L+IMAX
C
C     CELL IN FRONT =NN =L+IXMAX
C
C     N1 =FLAG AT THE LEFT
C     N2 =FLAG AT THE RIGHT
C     N3 =FLAG AT THE TOP
C     N4 =FLAG AT THE BOTTOM
C     N5 =FLAG AT THE BEHIND
C     N6 =FLAG AT THE IN FRONT
C
C     SET THE FLAGS FOR INCREASING OR DECREASING ACTIVE GRID
C        COUNTERS TO ZERO
      IX1=0
      IX2=0
      JY1=0
      JY2=0
      KZ1=0
      KZ2=0
C     SET FLAG FOR SUBCYCLING
      VEL=1.
C
C     RETURN HERE FOR THE SECOND PASS
C
C     SET UP THE K LOOP FIRST
    2 DO 3 K=K1,K2
    7 L=(K-1)*(IXMAX)+1
C
C     CHECK FOR BOUNDARY CONDITIONS AT THE LEFT
C
    8 DO 3302 J=1,JMAX
    6 IF(N1)9,99,9
C
C
C     TRANSMITTIVE LEFT BOUNDARY
C
   99 PL(J)=0.
      UL(J)=U(L)
      GO TO 10
C
C     REFLECTIVE  LEFT BOUNDARY
C
    9 PL(J)=P(L)
      UL(J)=0.
   10 L=L+IMAX
 3302 CONTINUE
   11 IF(K-1)8999,12,7001
```

```
 7001 IF(K-K1)15,12,15
C
C     BACK SIDE  BC. HAVE ALREADY BEEN SET
C
C
   12 CONTINUE
C     BACK SIDE  IS TRANS, BUT WILL TAKE CARE OF IT LATER
C     BACK SIDE  IS  REFLECTIVE
C
   13 DO 2302 N=1,IXMAX
      IF(N5)6010,6011,6010
 6011 PBIND(N)=0.
      UBIND(N)=W(N)
      GO TO 2302
 6010 PBIND(N)=P(N)
      UBIND(N)=0.
 2302 CONTINUE
   15 LL=(K-1)*IXMAX
C
C     SET  DO LOOP IN X DIRECTION
C
      DO 4 I=I1,I2
   16 M=(J1-1)*IMAX+I
   17 L=LL+I+(J1-1)*IMAX
C
C     SET  DO LOOP IN Y DIRECTION
C
   18 DO 5 J=J1,J2
      NN=L+IXMAX
   19 N=L+IMAX
   20 IF(J-1)9902,21,7003
 7003 IF(J1-J)3305,23,3305
C
C     WE HAVE ALREADY CALCULATED THE BOTTOM BC.
C     CHECK FOR BOTTOM BOUNDARY
   21 IF(N4)22,23,22
C
C     BOTTOM BOUNDARY IS TRANS
C
   23 VBLO=V(L)
      PBLO=0.
      GO TO 3305
C
C     BOTTOM BOUNDARY IS REFLECTIVE
C
   22 VBLO=0.
      PBLO=P(L)
C
C     NOW,WE HAVE THE LEFT BC.(IF REFLECTIVE)AND THE BOTTOM(IF REFLECT)
C
 3305 IF(AMX(L))9900,3340,3306
C
C     CELL IN QUESTION IS VOID, GET OUT AND CONTINUE THE LOOP
C
 3306 IF(IMAX-I)9901,3311,3307
C
C     WE ARE AT THE RIGHT BOUNDARY OF THE GRID
```

```
C
C       THE TOP IS REFLECTIVE
C
   30 PABOVE=P(L)
      VABOVE=0.
      GO TO 3328
C
C       THE TOP IS TRANS
C
   31 PABOVE=PBLO
C
C       MODIFY ETH FOR  TRANS BOUNDARY
C
      ETH=ETH-PABOVE/2.*V(L)*DT*DX(I)*DZ(K)
      GO TO 3323
C
C       WE ARE NOT AT THE TOP
 3320 IF(AMX(N))9905,3322,3324
C
C       CELL ABOVE IS VOID
 3322 PABOVE=0.
 3323 VABOVE=V(L)
      GO TO 3328
C
C       NORMAL FLOW FOR ALL CELLS OCCUPIED
 3324 PABOVE=(P(L)+P(N))/2.
   32 IF(1-J)3325,33,9906
C
C       BOTTOM BOUNDARY HAS BEEN SET
C       WE ARE AT THE BOTTOM
   33 IF(N4)3325,7000,3325
C
C        REFLECTIVE BOTTOM BOUNDARY CONDITION HAS ALREADY
C                BEEN SET.
C
 7000 PBLO=PABOVE
      ETH=ETH+PBLO/2.*V(L)*DT*DX(I)*DZ(K)
 3325 VABOVE=(V(L)+V(N))/2.
C
C       CHECK THE Z DIRECTION
C
 3328 IF(KMAX-K)9907,4418,4420
C
C       WE ARE AT THE FRONT OF THE GRID
C
 4418 IF(N6)2999,34,2999
C
C       FRONT IS  REFLECTIVE
 2999 PZR=P(L)
      WZR=0.
      GO TO 4328
```

```
C
C      FRONT IS  TRANS
   34 PZR=PBIND(M)
C
C      MODIFY  ETH FOR TRANS BOUNDARY
 4419 ETH=ETH-PZR/2.*W(L)*DT*DX(I)*DY(J)
      GO TO 4323
C
C      CHECK CELL  IN FRONT
C
 4420 IF(AMX(NN))9908,35,4324
C
C      CELL  IN FRONT   IS OCCUPIED
C      CELL  IN FRONT   IS VOID
C
   35 PZR=0.
 4323 WZR=W(L)
      GO TO 4328
C
C      NORMAL FLOW IN Z DIRECTION FOR ALL CELLS OCCUPIED
 4324 PZR=(P(L)+P(NN))/2.
      IF(1-K)4325,37,9909
C
C      BC. BEHIND HAVE ALREADY BEEN SET
C
C      WE ARE IN THE FIRST (X-Y )PLANE  K=1
   37 IF(N5)4325,8000,4325
C
C      REFLECTIVE BC. BEHIND HAVE ALREADY BEEN SET
C      TRANS BC. IN THE BACK
 8000 PBIND(M)=PZR
C
C      MODIFY ETH FOR TRANS BOUNDARY
      ETH=ETH+PZR/2.*W(L)*DT*DY(J)*DX(I)
 4325 WZR=(W(L)+W(NN))/2.
C
C      CHECK  FOR FIRST OR SECOND PASS
C
 4328 IF(VLL)9910,42,3400
C
C      THIS IS THE SECOND PASS,SKIP THE MOMENTA EQUATIONS
C
C
C
C      INTEGRATE THE  Y COMPONENT  OF VELOCITY (V)
 3400 V(L)=V(L)+(PBLO-PABOVE)/AMX(L)*DT*DX(I)*DZ(K)
```

```
C     CHECK THE MASS TO THE RIGHT
 3307 IF(AMX(L+1))9903,3312,3314
C
C     THIS IS THE BC. AT THE RIGHT OF A OCCUPIED CELL ,WITH THE
C     NEIGHBOR VOID.
C
 3312 PRR=0.
 3313 URR=U(L)
      GO TO 3316
C
C     HERE, WE ARE AT THE RIGHT BOUNDARY OF GRID (I=IMAX)
C     CHECK HERE FOR REFLECT OR TRANS BC.
C
 3311 IF(N2)3308,3309,3308
C
C     REFLECTIVE
C
 3308 PRR=P(L)
      URR=0.
      GO TO 3316
C
C     TRANSMITTIVE
C
 3309 PRR=PL(J)
C
C     MODIFY ETH HERE AT THE TRANS BOUNDARY
C
      ETH=ETH-PRR/2.*U(L)*DT*DY(J)*DZ(K)
      GO TO 3313
C
C     HERE IS NORMAL FLOW FOR ALL CELLS OCCUPIED
 3314 PRR=(P(L)+P(L+1))/2.
      URR=(U(L)+U(L+1))/2.
C
C     CHECK HERE FOR ALONG THE LEFT BOUNDARY (I=1) FOR TRANS
 3316 IF(I-1)9911,50,3310
   50 IF(N1)3310,51,3310
C
C     REFLECTIVE (BUT BC. HAVE ALREADY BEEN SET)
C
C     TRANSMITTIVE
   51 PL(J)=PRR
C
C     MODIFY ETH AT TRANS BOUNDARY
   52 ETH=ETH+PRR/2.*U(L)*DT*DY(J)*DZ(K)
      GO TO 3310
 3310 IF(JMAX-J)9904,3318,3320
C
C     WE ARE AT THE TOP OF THE GRID
 3318 IF(N3)30,31,30
```

```
      IF(ABS(V(L))-1.E-8)3401,3401,3402
 3401 V(L)=0.
C
C     INTEGRATE THE  X COMPONENT  OF VELOCITY (U)
 3402 U(L)=U(L)+(PL(J)-PRR)/AMX(L)*DT*DY(J)*DZ(K)
   40 IF(ABS(U(L))-1.E-8)3403,3403,3404
 3403 U(L)=0.
C
C     INTEGRATE THE  Z COMPONENT  OF VELOCITY (W)
 3404 W(L)=W(L)+(PBIND(M)-PZR)/AMX(L)*DT*DY(J)*DX(I)
   41 IF(ABS(W(L))-1.E-8)3405,3405,42
 3405 W(L)=0.
C
C
C     HERE CALCULATE THE CHANGE IN INTERNAL ENERGY DUE TO THE
C     WORK TERMS
   42 WS=P(L)*DT/AMX(L)*((VBLO-VABOVE)/2.*DX(I)*DZ(K)
     1+(UL(J)-URR)/2.*DY(J)*DZ(K)
     2+(UBIND(M)-WZR)/2.*DX(I)*DY(J))
   43 AIX(L)=AIX(L)+WS
C
C      CHECK ON ADVANCING OR DECREASING GRID COUNTERS
C
 5800 IF(I-I2)5999,5801,5801
 5801 IF(IX2)5999,5802,5999
 5802 IF(ABS(U(L))+ABS(V(L))+ABS(W(L))+AIX(L))5803,5999,5803
 5803 IX2=1
      GO TO 5999
 5999 IF(K-K2)4999,5804,5804
 5804 IF(KZ2)4999,5805,4999
 5805 IF(ABS(U(L))+ABS(V(L))+ABS(W(L))+AIX(L))5806,4999,5806
 5806 KZ2=1
      GO TO 4999
 4999 IF(KZ1)5300,5222,5300
 5222 IF(K1-1)5300,5300,5000
 5000 IF(K1-K)5300,5001,5001
 5001 IF(ABS(U(L))+ABS(V(L))+ABS(W(L))+AIX(L))5002,5300,5002
 5002 KZ1=1
 5300 IF(JY1)5600,5304,5600
 5304 IF(J1-1)5600,5600,5301
 5301 IF(J1-J)5600,5302,5302
 5302 IF(ABS(U(L))+ABS(V(L))+ABS(W(L))+AIX(L))5303,5600,5303
 5303 JY1=1
 5600 IF(IX1)3342,5604,3342
 5604 IF(I1-1)3342,3342,5601
 5601 IF(I1-I)3342,5602,5602
 5602 IF(ABS(U(L))+ABS(V(L))+ABS(W(L))+AIX(L))5603,3342,5603
 5603 IX1=1
      GO TO 3342
C
C     CAME HERE BECAUSE THE CELL IN QUESTION (L) IS VOID
C
 3340 PRR=0.
      URR=U(L+1)
      PABOVE=0.
      VABOVE=V(N)
      PZR=0.
```

```
      WZR=W(NN)
C
C     SET THE ABOVE QUANTITIES TO BELOW
 3342 VBLO=VABOVE
      PBLO=PABOVE
C
C     SET THE RIGHT QUANTITIES TO THE LEFT
      PL(J)=PRR
      UL(J)=URR
C
C     SET THE FRONT QUANTITIES TO BEHIND
      PBIND(M)=PZR
      UBIND(M)=WZR
C
C     UPDATE THE INDICES
C
      L=N
      M= M+IMAX
C     TERMINATION OF LOOP ON J---(Y)
    5 CONTINUE
C      CHECK ON ADVANCING OR DECREASING GRID COUNTERS
 5700 LJ=L-IMAX
      IF(JY2)4,5701,4
 5701 IF(ABS(U(LJ))+ABS(V(LJ))+ABS(W(LJ))+AIX(LJ))5702,4,5702
 5702 JY2=1
      GO TO 4
C
C     TERMINATION OF LOOP ON I---(X)
    4 CONTINUE
C
C     TERMINATION OF LOOP ON K---(Z)
    3 CONTINUE
C
C
C     CHECK FOR FIRST OR SECOND PASS
   44 IF(VEL-1.)46,45,46
   45 VEL=0.
C
C     RECYCLE
      GO TO 2
C
C     HAVE COMPLETED BOTH PASSES
   46 CONTINUE
C
C      INCREASE OR DECREASE COUNTERS AS REQURED
C
      I1=I1-IX1
 5900 IF(I1)5901,5901,5902
 5901 I1=1
 5902 I2=I2+IX2
 5903 IF(I2-IMAX)5905,5905,5904
 5904 I2=IMAX
 5905 J1=J1-JY1
      IF(J1)5906,5906,5907
 5906 J1=1
 5907 J2=J2+JY2
      IF(J2-JMAX)5909,5909,5908
```

```
 5908 J2=JMAX
 5909 K1=K1-KZ1
      IF(K1)5910,5910,5911
 5910 K1=1
 5911 K2=K2+KZ2
      IF(K2-KMAX)5913,5913,5912
 5912 K2=KMAX
 5913 RETURN
 8999 NK=11
      GO TO 9999
 9902 NK=20
      GO TO 9999
 9900 NK=3305
      GO TO 9999
 9901 NK=3306
      GO TO 9999
 9903 NK=3307
      GO TO 9999
 9911 NK=3316
      GO TO 9999
 9904 NK=3310
      GO TO 9999
 9905 NK=3320
      GO TO 9999
 9906 NK=32
      GO TO 9999
 9907 NK=3328
      GO TO 9999
 9908 NK=4420
      GO TO 9999
 9909 NK=4324
      GO TO 9999
 9910 NK=4328
 9999 NR=3
      WRITE(6,8500)I,J,K,L,M,N,NN,NK,NR
 8500 FORMAT(9I6)
      CALL UNCLE
C     CALL DUMP
      CALL DUMP
      END
@I FOR   PH2/S,PH2/S,PH2/SS
      SUBROUTINE PH2                                                PH2 0010
      DIMENSION  AIX(6000),AMX(6000),U(6000),V(6000),W(6000),P(6000),
C
C     ********* 3DOIL ****************************
C
C     HERE,WE APPROXIMATE  THE  TRANSPORT  TERMS LEFT OUT OF THE
C          MOMENTUM AND ENERGY EQUATIONS  IN PH1  BY  MOVING
C          MASS,(SOLVING THE MASS CONSERVATION EQUATION) THIS MASS
C          THEN CARRIES ENERGY AND  MOMENTUM  ACROSS THE FIXED
C          GRID LINES
C
C     AMPY = MASS FLOW  AT THE TOP
C     AMUT = X MOMENTUM COMPONENT OF THIS MASS
C     AMVT = Y MOMENTUM COMPONENT OF THIS MASS
C     AMWT = Z MOMENTUM COMPONENT OF THIS MASS
C     DELET= SPECIFIC ENERGY OF THIS MASS
```

```
C
C     AMMP = MASS FLOW AT THE RIGHT
C     AMUR = X MOMENTUM COMPONENT OF THIS MASS
C     AMVR = Y MOMENTUM COMPONENT OF THIS MASS
C     AMWR = Z MOMENTUM COMPONENT OF THIS MASS
C     DELER= SPECIFIC ENERGY OF THIS MASS
C
C     AMMY= MASS FLOW AT THE BOTTOM
C     AMMU= X MOMENTUM COMPONENT OF  THIS MASS
C     AMMV= Y MOMENTUM COMPONENT OF  THIS MASS
C     AMMW= Z MOMENTUM COMPONENT OF  THIS MASS
C     DELEB= SPECIFIC ENERGY  OF  THIS MASS
C
C     GAMC = MASS FLOW AT THE LEFT
C     FLEFT= X MOMENTUM COMPONENT OF THIS MASS
C     YAMC = Y MOMENTUM COMPONENT OF THIS MASS
C     ZMOM = W MOMENTUM COMPONENT OF THIS MASS
C     SIGC = SPECIFIC ENERGY OF  THIS MASS
C
C     BMASS= MASS FLOW  AT THE BACK
C     BXMOM= X MOMENTUM COMPONENT OF THIS MASS
C     BYMOM= Y MOMENTUM COMPONENT OF THIS MASS
C     BZMOM= Z MOMENTUM COMPONENT OF THIS MASS
C     BENR = SPECIFIC ENERGY OF THIS  MASS
C
C     FMASS= MASS FLOW IN FRONT
C     FXMOM= X MOMENTUM COMPONENT OF THIS MASS
C     FYMOM= Y MOMENTUM COMPONENT OF THIS MASS
C     FZMOM= Z MOMENTUM COMPONENT OF THIS MASS
C     FENR = SPECIFIC ENERGY OF THIS MASS
C
      REZ=0.
C
C     INITIALIZE THE FLAGS FOR ADVANCING THE
C      ACTIVE GRID COUNTERS TO ZERO
      IX1=0
      IX2=0
      JY1=0
      JY2=0
      KZ1=0
      KZ2=0
      SUM=0.
      CALL SLITE(0)
C
C     DO LOOP  ON K
C
    5 DO 1 K=K1,K2
      LL=(K-1)*IXMAX
C
C     DO LOOP  ON I
    6 DO 2 I=I1,I2
C
C     NOTE (L) IS  THE  CELL  INDEX  =(J-1)IMAX + I
C         + (K-1)IXMAX   (NOTE  IXMAX=(IMAX)(JMAX)
C
      L=LL+I+(J1-1)*IMAX
C
```

```
C      DO LOOP  ON J
C
     7 DO 3 J=J1,J2
C      NN = THE INDEX OF THE CELL IN FRONT = L+ IXMAX
       NN=L+IXMAX
C
C      N= THE INDEX OF THE CELL ABOVE  = L+ IMAX
       N=L+IMAX
C
C      M= INDEX OF THE CELL IN QUESTION FOR A SINGLE PLANE
C
       M=(J-1)*IMAX+I
C      N1,N2,N3,N4,N5,N6  ARE  FLAGS  TO SET  BOUNDARY CONDITIONS
C      AT THE  6 FACES  OF  THIS GRID
C
C      N1 REFERS TO THE LEFT
C      N2 REFERS TO THE RIGHT
C      N3 REFERS TO THE TOP
C      N4 REFERS TO THE BOTTOM
C      N5 REFERS TO BEHIND
C      N6 REFERS TO IN FRONT
C
C      FREE SURFACES  WITHIN THE GRID ARE  TREATED AS FOLLOWS ,
C          IF THE MASS FLOW  INTO  A EMPTY CELL  PRODUCES A
C          DENSITY THAT IS LESS THAN  TOZONE(A INPUT NUMBER LIKE
C          .001 OF RHONOT),THE MASS FLUX IS SET TO ZERO
C
   600 IF(J-1)9903,601,9302
  9302 IF(J1-J)9,603,9
C
C      BOTTOM BC. HAS BEEN SET
C
   601 IF(AMX(L))9904,603,602
C
C      WE ARE AT THE BOTTOM OF THE  X-Y PLANE
   602 IF(V(L))604,603,603
C
C      SET Y COMPONENT OF MOMENTUM TO 0.
   603 AMMV=0.
       GO TO 698
C
C      CALCULATE THE MASS FLUX AT THE BOTTOM
   604 AMMY=AMX(L)/DY(J)*V(L)*DT
C
C      CHECK SO MASS FLUX DOES NOT MORE THAN EMPTY THE CELL
   605 IF(AMMY+AMX(L))9300,607,607
  9300 AMMY=-AMX(L)
   607 IF(N4)609,608,609
C
C      BOTTOM BOUNDARY IS TRANS
   608 AMMU=AMMY*U(L)
C
C      CALCULATE THE 3 MOMENTAS,THE ENERGY
C      SUBTRACT THIS ENERGY LOSS FROM ETH
       AMMV=AMMY*V(L)
       AMMW=AMMY*W(L)
       WS=(U(L)**2+V(L)**2+W(L)**2)/2.
```

```
      DELEB=AIX(L)+WS
      ETH=ETH+AMMY*DELEB
      IF(-AMMY/(DX(I)*DY(J)*DZ(K))-Z(80))610,610,6600
 6600 REZ=1.0
      GO TO 610
C
C     BOTTOM BOUNDARY IS REFLECTIVE, NET MOMENTA CHANGE= 2MV
  609 AMMV=2.*AMMY*V(L)
C
C     SET MASS, X AND Z MOMENTA AND  SPECIFIC ENERGY TO 0.
  698 AMMY=0.
      AMMU=0.
      AMMW=0.
      DELEB=0.
  610 CONTINUE
C
C     *** FINISHED WITH THE BOTTOM BC. ***********
C     *****************************************************************
    9 IF(I-1)8999,10,9301
 9301 IF(I1-I)506,9310,506
 9310 JJ=J
      GO TO 20
C
C     WE ARE ALONG THE LEFT BOUNDARY ( I=1)
   10 NL=L
   11 JJ=J
   15 IF(AMX(NL))9900,20,16
   20 FLEFT(JJ)=0.
      GO TO 5504
   16 IF(U(NL))17,20,20
C
C     CALCULATE MASS FLUX
   17 GAMC(JJ)=AMX(NL)/DX(1)*U(NL)*DT
   21 IF(GAMC(JJ)+AMX(NL))22,500,500
   22 GAMC(JJ)=-AMX(NL)
  500 IF(N1)501,502,501
C
C     LEFT BOUNDARY IS TRANS
C     CALCULATE THE 3 MOMENTAS,THE ENERGY
C     SUBTRACT THIS ENERGY LOSS FROM ETH
  502 FLEFT(JJ)=GAMC(JJ)*U(NL)
      YAMC(JJ)=GAMC(JJ)*V(NL)
      ZMOM(JJ)=GAMC(JJ)*W(NL)
      WS=(U(NL)**2+V(NL)**2+W(NL)**2)/2.
      SIGC(JJ)=AIX(NL)+WS
      ETH=ETH+GAMC(JJ)*SIGC(JJ)
      IF(-GAMC(J)/(DX(I)*DY(J)*DZ(K))-Z(75))503,503,6610
 6610 REZ=1.0
      GO TO 503
C
C     LEFT BOUNDARY IS REFLECTIVE, NET MOMENTA CHANGE =2MU
  501 FLEFT(JJ)=2.*GAMC(JJ)*U(NL)
C
C     SET MASS, Y AND Z MOMENTA AND SPECIFIC ENERGY TO 0.
 5504 GAMC(JJ)=0.
      YAMC(JJ)=0.
      ZMOM(JJ)=0.
```

```
      SIGC(JJ)=0.
  503 CONTINUE
  103 CONTINUE
C
C     FINISHED WITH LEFT BOUNDARY CONDITIONS
C
C     ******************************************************************
C
  506 IF(K-1)9901,23,9303
 9303 IF(K1-K)31,250,31
   23 IF(AMX(L))9902,250,24
  250 BZMOM(M)=0.
      GO TO 25
C
C     CHECK Z COMPONENT OF VELOCITY
   24 IF(W(L))26,250,250
C
C     SET THE 5 DATA BEHIND TO 0.
   25 BMASS(M)=0.
      BXMOM(M)=0.
      BYMOM(M)=0.
      BENR(M)=0.
      GO TO 31
C
C     VELOCITY IS - ,CALCULATE THE MASS FLUX
   26 BMASS(M)=AMX(L)/DZ(K)*W(L)*DT
C
C     CHECK SO WE DONT EMPTY MORE MASS THAN THERE IS
   27 IF(BMASS(M)+AMX(L))28,40,40
   28 BMASS(M)=-AMX(L)
C
C     CHECK FOR TRANS OR REFLECT
   40 IF(N5)41,29,41
C
C     REFLECTIVE
   41 BZMOM(M)=2.*BMASS(M)*W(L)
      GO TO 25
C
C     TRANSMITTIVE
C     CALCULATE  THE MOMENTAS OF THIS MASS
   29 BXMOM(M)=BMASS(M)*U(L)
      BYMOM(M)=BMASS(M)*V(L)
      BZMOM(M)=BMASS(M)*W(L)
C
C     CALCULATE  THE TOTAL ENERGY CARRIED BY THIS MASS
   30 WS=(U(L)**2+V(L)**2+W(L)**2)/2.
      BENR(M)=AIX(L)+WS
C
C     REMOVE THE ENERGY LOSS FROM ETH
      ETH=ETH+BMASS(M)*BENR(M)
      IF(-BMASS(M)/(DX(I)*DY(J)*DZ(K))-Z(78))31,31,6620
 6620 REZ=1.0
C
C     HAVE CALCULATED THE DATA BEHIND, NOW CHECK ON JMAX
C
C     ****************************************************
C      NOW, UP TO THIS POINT, WE HAVE TAKEN CARE OF
```

```
C         BOTH REFLECTIVE AND TRANSMITTIVE BOUNDARIES
C       AT THE BOTTOM, LEFT, AND BACK
C       ***************************************************
C       ****************************************************************
C
C
C       VABOVE CALC.
C       *******************************************
   31 VEL=0.
C
C
C       SET UP TO  CALCULATE VABOVE
        IF(JMAX-J)211,211,207
C
C       WE ARE AT THE TOP OF THE GRID
  211 VEL=1.
        GO TO 208
C
C       CHECK CELL ABOVE
  207 IF(AMX(N))215,215,214
C
C       CELL ABOVE IS VOID
  214 IF(AMX(L))216,216,209
C
C       CELL (L) IS VOID, BUT CELL  ABOVE IS OCCUPIED
  216 VABOVE=V(N)
        GO TO 212
  215 IF(AMX(L))205,205,208
C
C       BOTH CELLS ARE VOID
C
  205 VABOVE=0.
        GO TO 212
  208 VABOVE=V(L)
        GO TO 212
C
C       BOTH CELLS ARE OCCUPIED
  209 VABOVE=(V(L)+V(N))/2.
  212 FS=0.
C
C
C       U RIGHT CALC.
C       *******************************************
C
C       NOW,BEGIN CALCULATION OF URR
C
  404 IF(IMAX-I)412,412,405
  405 IF(AMX(L+1))411,411,409
  409 IF(AMX(L))410,410,407

C       CELL (L) IS VOID ,BUT CELL TO THE RIGHT IS FILLED
  410 URR=U(L+1)
        GO TO 408
C
C       MASS ON THE RIGHT=0.
  411 IF(AMX(L))403,403,406
  403 URR=0.
```

```
      GO TO 408
C
C     WE ARE AT THE RIGHT OF THE GRID
  412 FS=1.
  406 URR=U(L)
      GO TO 408
C
C     THIS IS THE NORMAL PATH
C     BOTH CELLS ARE FILLED
  407 URR=(U(L)+U(L+1))/2.
  408 CONTINUE
C
C     W IN FRONT CALC.
C     ********************************************
C
C     NOW ,WE HAVE  VABOVE (V AT THE TOP) AND URR(V AT THE RIGHT)
 5503 AREA=0.
C
C     LETS CALCULATE  WOUT (THE Z COMPONENT)
  504 IF(KMAX-K)512,512,505
  505 IF(AMX(NN))511,511,509
  509 IF(AMX(L))510,510,507
C
C     CELL (L) IS VOID ,BUT CELL IN FRONT IS  OCCUPIED
  510 WOUT=W(NN)
      GO TO 508
C
C     MASS IN FRONT IS 0.
  511 IF(AMX(L))5513,5513,5066
 5513 WOUT=0.
      GO TO 508
C
C     WE ARE AT THE  FRONT BOUNDARY
  512 AREA=1.
 5066 WOUT=W(L)
      GO TO 508
C
C     NORMAL PATH
C     BOTH CELLS ARE FILLED
  507 WOUT=(W(L)+W(NN))/2.
C
C     IF VEL IS GREATER THAN 0. ,WE ARE IN THE TOP CELL (J=JMAX)
C     IF FS IS GREATER THAN 0., WE ARE IN THE  RIGHT COLUMN
C       THAT IS     (I=IMAX)
C     IF AREA IS GREATER THAN 0., WE ARE  IN THE FRONT SLAB(X-YPLANE)
C       THAT IS     (K=KMAX)
  508 CONTINUE
C
C     ***********************************************
C     NOW, FINALLY, WE HAVE ALL 3 INTERFACE VELOCITIES
C        *  *  *  *  *  *  *  *  *  *  *  *  *  *
C     HERE WE BEGIN THE CALCULATION OF THE 3 FLUXES
C
C     ********************************************
C     IF TOP IS TRANS. THE ENERGY LOSS IS CALCULATED LATER
C     ********************************************
  100 IF(VABOVE)102,101,1103
```

```
C
C      Y FLUX IS  POSITIVE
 1103 IF(AMX(L))9910,101,104
C
C      SET INDICES
  104 IF(J-1)9910,6104,6105
 6105 KP=L-IMAX
      IF(AMX(KP))9910,6106,6104
 6106 IF(ABS(VABOVE-VELOC)/VELOC-BUG)107,6104,6104
 6104 LY=L
      JY=J
      IF(VEL)105,105,109
C
C      Y FLUX = 0.
  101 AMPY=0.
      AMUT=0.
      AMVT=0.
      AMWT=0.
      DELET=0.
      GO TO 115
C
C      Y FLUX IS NEGATIVE (DOWN)
C      FLUX IS NEGATIVE FROM CELL (L)
  102 IF(VEL)106,106,101
C
C      FLUX IS - BUT FROM THE TOP CELL
C      CHECK CELL ABOVE
  106 IF(AMX(N))9911,101,107
C
C      FLUX IS NEGATIVE,BUT CELL MASS =0.
  107 LY=N
      JY=J+1
  105 VABOVE= (VABOVE)/(1.+(V(N)-V(L))/DY(JY)*DT/SBOUND)
C
C      CALCULATE FLUX AT THE TOP
C        * * * * * * * * * * *
  109 AMPY=AMX(LY)*VABOVE/DY(JY)*DT
  110 IF(VEL)115,115,111
C
C      WE ARE AT THE TOP,CHECK THE BC. AT THE TOP
  111 IF(N3)112,115,112
C
C      REFLECT THE MASS
  112 IF(AMPY)115,115,113
  113 AMVT=-2.*AMPY*V(L)
  114 AMPY=0.
      AMUT=0.
      AMWT=0.
      DELET=0.
C      ********************************************
C       IF RIGHT IS TRANS. THE ENERGY LOSS IS CALCULATED LATER.
C      ********************************************
C
C      BEGIN CALCULATING THE FLUX AT THE RIGHT
  115 IF(URR)118,116,117
  116 AMMP=0.
      AMUR=0.
```

```
      AMVR=0.
      AMWR=0.
      DELER=0.
C
C
C     CALCULATE THE MASSFLOW IN THE Z DIRECTION
      GO TO 300
  117 IF(AMX(L))9911,116,120
  120 LX=L
      IX=I
      IF(FS)119,119,122
C
C     RIGHT FLUX IS NEGATIVE
  118 IF(FS)130,130,116
C
C     FLUX IS NEGATIVE ,BUT CELL MASS IS ZERO ALSO
C
C     CHECK THE CELL TO THE RIGHT
  130 IF(AMX(L+1))9911,116,121
  121 LX=L+1
      IX=I+1
  119 WS=(U(L+1)-U(L))/DX(IX)*DT/SBOUND
      URR=URR/(1.+WS)
C
C     CALCULATE  THE MASS FLUX AT THE RIGHT
C      * * * * * * * * * * * * * * * * *
  122 AMMP=AMX(LX)/DX(IX)*URR*DT
      IF(FS)123,300,123
C
C     CHECK THE BOUNDARY CONDITION
  123 IF(N2)124,300,124
C
C     TRANS
  124 IF(AMMP)300,300,125
C
C     REFLECT THE MASS
  125 AMUR=-2.*AMMP*U(L)
      AMMP=0.
      AMVR=0.
      AMWR=0.
      DELER=0.
C
C     IF FRONT IS TRANS. THE ENERGY LOSS IS TAKEN CARE OF LATER.
C
C     DO THE Z COMPONENT NOW
C
C     SET THE 5 DATA IN FRONT TO  0.
  300 IF(WOUT)318,316,317
  316 FMASS=0.
      FXMOM=0.
      FYMOM=0.
      FZMOM=0.
      FENR=0.
      GO TO 700
  317 IF(AMX(L))9912,316,3200
 3200 LZ=L
      IZZ=K
```

```
      IF(AREA)319,319,322
C
C     FRONT FLUX IS NEGATIVE
  318 IF(AREA)320,320,316
C
C     FLUX IS NEGATIVE, BUT FROM IN FRONT
C     CHECK CELL IN FRONT
  320 IF(AMX(NN))9912,316,321
C
C     FLUX IS NEGATIVE ,BUT CELL MASS =0.
  321 LZ=NN
      IZZ=K+1
  319 WS=(W(NN)-W(L))/DZ(IZZ)*DT/SBOUND
      WOUT=WOUT/(1.+WS)
C
C     CALCULATE THE MASS FLUX IN FRONT
C      * * * * * * * * * * * * *  * *
  322 FMASS=AMX(LZ)/DZ(IZZ)*WOUT*DT
      IF(AREA)323,700,323
  323 IF(N6)324,700,324
C
C     TRANS
C
C     REFLECT,WE ARE IN FRONT
  324 IF(FMASS)700,700,325
C
C     REFLECT THE MASS
  325 FZMOM=-2.*FMASS*W(L)
      FMASS=0.
      FXMOM=0.
      FYMOM=0.
      FENR=0.
C
C     ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
C      NOW WE HAVE ALL 3 FLUXES AND ALL
C       THE BOUNDARY CONDITIONS HAVE BEEN SET
  700 IF(AMPY)760,980,761
C
C     TOP FLUX IS -
  760 IF(AMPY+AMX(N))762,980,980
  762 AMPY=-AMX(N)
      GO TO 980
  761 IF(-AMPY+AMX(L))763,980,980
  763 AMPY=AMX(L)
  980 IF(AMMP)7300,981,7301
 7300 IF(AMMP+AMX(L+1))7302,981,981
 7302 AMMP=-AMX(L+1)
      GO TO 981
 7301 IF(-AMMP+AMX(L))7303,981,981
 7303 AMMP=AMX(L)
  981 IF(FMASS)7400,982,7401
 7400 IF(FMASS+AMX(NN))7402,982,982
 7402 FMASS=-AMX(NN)
      GO TO 982
 7401 IF(-FMASS+AMX(L))7403,982,982
 7403 FMASS=AMX(L)
  982 WS=GAMC(J)
```

```
      IF(WS)902,901,901
  901 WS=0.
  902 WSA=AMMY
      IF(WSA)904,903,903
  903 WSA=0.
  904 WSB=BMASS(M)
      IF(WSB)906,905,905
  905 WSB=0.
  906 WSC=AMX(L)+WS+WSA+WSB
  907 WS=AMPY
      IF(WS)950,950,909
  950 WS1=0.
      IF(K-K1)951,953,951
  951 IF(GAMC(J+1))952,953,953
  952 WS1=GAMC(J+1)
  953 WS2=0.
      NA=M+IMAX
      IF(K-K1)955,957,955
  955 IF(BMASS(NA))956,957,957
  956 WS2=BMASS(NA)
  957 WS3=WS1+WS2+AMX(N)
  958 IF(AMPY+WS3)959,908,908
  959 AMPY=-WS3
      GO TO 908
  908 WS=0.
  909 WSA=AMMP
      IF(WSA)970,970,911
  970 WS1=0.
      IF(K-K1)971,973,971
  971 IF(BMASS(M+1))972,973,973
  972 WS1=BMASS(M+1)
  973 WS3=WS1+AMX(L+1)
  974 IF(AMMP+WS3)975,910,910
  975 AMMP=-WS3
      GO TO 910
  910 WSA=0.
  911 WSB=FMASS
      IF(WSB)912,912,913
  912 WSB=0.
  913 WST=WS+WSA+WSB
      IF(WST)921,921,931
  931 IF(WSC)932,932,933
  932 IF(AMPY)934,934,935
  935 AMPY=0.
  934 IF(AMMP)936,936,937
  937 AMMP=0.
  936 IF(FMASS)921,921,938
  938 FMASS=0.
      GO TO 921
  933 IF(WSC-WST)914,921,921
  914 WSD=WSC/WST
      WS=WS*WSD
      WSA=WSA*WSD
      WSB=WSB*WSD
  915 IF(WS)917,917,916
  916 AMPY=WS
  917 IF(WSA)919,919,918
```

```
  918 AMMP=WSA
  919 IF(WSB)921,921,920
  920 FMASS=WSB
  921 CONTINUE
      IF(AMPY)703,2700,702
 2700 IF(J-JMAX)717,2701,2701
 2701 IF(N3)716,717,716
  717 AMUT=0.
      AMVT=0.
      AMWT=0.
      DELET=0.
      GO TO 716
C
C     TOP FLUX IS +
  702 IF(JMAX-J)9913,704,705
C
C     CHECK CELL ABOVE
  705 IF(AMX(N))9914,706,704
C
C     FREE  SURFACE AT TOP
  706 IF(AMPY/(DX(I)*DY(J)*DZ(K))-TOZONE)707,704,704
C
C     DENSITY IS  TOO SMALL, SET FLUX =0.
  707 AMPY=0.
      GO TO 717
C
C     TOP FLUX IS NEGATIVE
  703 IF(J-1)9913,701,709
  709 IF(AMX(L))9914,710,701
  710 IF(-AMPY/(DX(I)*DY(J)*DZ(K))-TOZONE)711,701,701
  711 AMPY=0.
      GO TO 717
C
C     ADD UP THE MASSES (REMEMBER THEY HAVE DIRECTION)
C
  704 DELM=GAMC(J)+AMMY+BMASS(M)-AMPY
      IF(VEL)9914,712,720
C     AT TOP OF GRID
  720 IF(N3)713,714,713
C
C     TOP BOUNDARY IS TRANSMITTIVE
  714 WS= U(L)**2+V(L)**2+W(L)**2
      ETH=ETH-AMPY*(AIX(L)+WS/2.)
      IF(AMPY/(DX(I)*DY(J)*DZ(K))-Z(79))712,712,6630
 6630 REZ=1.
C
C     CALCULATE THE MOMENTUMS
  712 AMUT=AMPY*U(L)
      AMVT=AMPY*V(L)
      AMWT=AMPY*W(L)
      GO TO 713
  701 AMUT=AMPY*U(N)
      AMVT=AMPY*V(N)
      AMWT=AMPY*W(N)
      DELET=AIX(N)+(U(N)**2+V(N)**2+W(N)**2)/2.
  716 DELM=GAMC(J)+AMMY+BMASS(M)-AMPY
      GO TO 715
```

```
 713 DELET=AIX(L)+(U(L)**2+V(L)**2+W(L)**2)/2.
 715 SIGMU=FLEFT(J)+AMMU-AMUT+BXMOM(M)
     SIGMV=YAMC(J)+AMMV-AMVT+BYMOM(M)
     DELEK=GAMC(J)*SIGC(J)+AMMY*DELEB-AMPY*DELET+BMASS(M)*BENR(M)
     SIGMW=ZMOM(J)+AMMW-AMWT+BZMOM(M)
     GO TO 7000
C
C    NOW,DO THE SAME FOR THE X DIRECTION
C    MASS FLUX AT THE RIGHT
C
C    ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
7000 IF(AMMP)7003,2702,7002
C
2702 IF(I-IMAX)7017,2703,2703
2703 IF(N2)7016,7017,7016
C
C    FLUX IS POSITIVE
7002 IF(IMAX-I)9914,7004,7005
7017 AMUR=0.
     AMVR=0.
     AMWR=0.
     DELER=0.
     GO TO 7100
C
C    CHECK CELL TO THE RIGHT
7005 IF(AMX(L+1))9915,7006,7004
C    FREE SURFACE AT THE RIGHT
7006 IF(AMMP/(DX(I)*DY(J)*DZ(K))-TOZONE)7007,7004,7004
C
C    DENSITY IS TOO SMALL, SET FLUX TO 0.
7007 AMMP=0.
     GO TO 7017
C
C    RIGHT FLUX IS NEGATIVE
7003 IF(I-1)9914,7001,7009
7009 IF(AMX(L))9915,7010,7001
7010 IF(-AMMP/(DX(I)*DY(J)*DZ(K))-TOZONE)7011,7001,7001
C
C    FLUX IS TOO SMALL
7011 AMMP=0.
     GO TO 7017
7004 DELM=DELM-AMMP
     IF(FS)9915,7012,7020
C
C    AT RIGHT BOUNDARY
7020 IF(N2)7150,7014,7150
C
C    RIGHT BOUNDARY IS TRANSMITTIVE
7014 WS=U(L)**2+V(L)**2+W(L)**2
     ETH=ETH-AMMP*(AIX(L)+WS/2.)
     IF(AMMP/(DX(I)*DY(J)*DZ(K)-Z(77))7012,7012,6640
6640 REZ=1.
7012 AMUR=AMMP*U(L)
     AMVR=AMMP*V(L)
     AMWR=AMMP*W(L)
     DELER=AIX(L)+(U(L)**2+V(L)**2+W(L)**2)/2.
     GO TO 7150
```

```
 7001 AMUR=AMMP*U(L+1)
      AMVR=AMMP*V(L+1)
      AMWR=AMMP*W(L+1)
      DELER=AIX(L+1)+(U(L+1)**2+V(L+1)**2+W(L+1)**2)/2.
 7016 DELM=DELM-AMMP
C
C      SUM UP TOTAL MOMENTA
 7150 SIGMU=SIGMU-AMUR
      SIGMV=SIGMV-AMVR
      SIGMW=SIGMW-AMWR
      DELEK=DELEK-AMMP*DELER
C
C      DO THE SAME FOR THE FLUX IN FRONT
C      ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
C
 7100 IF(FMASS)7103,2704,9982
 2704 IF(K-KMAX)7117,2705,2705
 2705 IF(N6)4000,7117,4000
C
C      FLUX IS POSITIVE
 9982 IF(KMAX-K)9916,7104,7105
 7105 IF(AMX(NN))9917,7106,7104
C
C      FREE SURFACE IN FRONT
 7106 IF(FMASS/(DX(I)*DY(J)*DZ(K))-TOZONE)7107,7104,7104
C
C      DENSITY IS TOO SMALL, SET MASS =0.
 7107 FMASS=0.
 7117 FXMOM=0.
      FYMOM=0.
      FZMOM=0.
      FENR=0.
      GO TO 4000
C
C      FRONT FLUX IS -
 7103 IF(K-1)9916,7101,7109
 7109 IF(AMX(L))9917,7110,7101
 7110 IF(-FMASS/(DX(I)*DY(J)*DZ(K))-TOZONE)7111,7101,7101
C
C      FLUX IS TOO SMALL
 7111 FMASS=0.
      GO TO 7117
 7104 DELM=DELM-FMASS+AMX(L)
      IF(AREA)9916,7112,7120
 7120 IF(N6)8000,7114,8000
C
C        FRONT BOUNDARY IS TRANSMITTIVE
 7114 WS=U(L)**2+V(L)**2+W(L)**2
      ETH=ETH-FMASS*(AIX(L)+WS/2.)
      IF(FMASS/(DX(I)*DY(J)*DZ(K))-Z(76))7112,7112,6650
 6650 REZ=1.
 7112 FXMOM=FMASS*U(L)
      FYMOM=FMASS*V(L)
      FZMOM=FMASS*W(L)
      FENR=AIX(L)+(U(L)**2+V(L)**2+W(L)**2)/2.
      GO TO 8000
 7101 FXMOM=FMASS*U(NN)
```

```
      FYMOM=FMASS*V(NN)
      FZMOM=FMASS*W(NN)
 7116 FENR=AIX(NN)+(U(NN)**2+V(NN)**2+W(NN)**2)/2.
 4000 DELM=DELM-FMASS+AMX(L)
 8000 SIGMU=SIGMU-FXMOM
      SIGMV=SIGMV-FYMOM
      SIGMW=SIGMW-FZMOM
      DELEK=DELEK-FMASS*FENR
C
C     TOTAL MASS AT CYCLE  N+1
C       * * * * * * * * * *
      IF(DELM)544,545,540
  544 IF(AMX(L)*1.E-6+DELM)9918,545,545
  545 DELM=0.
      GO TO 550
  540 WS=U(L)**2+V(L)**2+W(L)**2
      WS=WS/2.
      ENK=AMX(L)*(AIX(L)+WS)+DELEK
      GO TO 541
C
C     HERE ,WE  CALCULATE  THE  3  NEW  CELL (L) VELOCITIES BY
C          CONSERVING MOMENTUM,  WE ALSO  CONSERVE THE TOTAL
C          ENERGY, THE  TOTAL ENERGY LESS THE KINETIC  IS
C          THAN THE NEW SPECIFIC INTERNAL ENERGY.
C
C     NEW X VEL COMPONENT
  541 U(L)=(SIGMU+AMX(L)*U(L))/DELM
C
C     NEW Y VEL COMPONENT
  546 V(L)=(SIGMV+AMX(L)*V(L))/DELM
C
C     NEW Z VEL COMPONENT
  547 W(L)=(SIGMW+AMX(L)*W(L))/DELM
  548 WS=U(L)**2+V(L)**2+W(L)**2
C
C     NEW SPECIFIC INTERNAL ENERGY
  549 AIX(L)=ENK/DELM-WS/2.
      IF(AIX(L)-TMASS)7500,7500,7510
 7500 SUM=SUM+AIX(L)*DELM
      AIX(L)=0.
 7510 IF(ABS(U(L))-XMAX)7501,7501,7502
 7501 WS=U(L)**2
      SUM=SUM+DELM*WS/2.
      U(L)=0.
 7502 IF(ABS(V(L))-XMAX)7503,7503,7504
 7503 WS=V(L)**2
      SUM=SUM+DELM*WS/2.
      V(L)=0.
 7504 IF(ABS(W(L))-XMAX)7505,7505,7506
 7505 WS=W(L)**2
      SUM=SUM+DELM*WS/2.
      W(L)=0.
 7506 IF(AIX(L))4001,550,550
 4001 SUM=SUM+AIX(L)*DELM
      AIX(L)=0.
  550 AMX(L)=DELM
 5800 IF(I-I2)5999,5801,5801
```

```
 5801 IF(IX2)5999,5802,5999
 5802 IF(ABS(U(L))+ABS(V(L))+ABS(W(L))+AIX(L))5803,5999,5803
 5803 IX2=1
      GO TO 5999
 5999 IF(K-K2)4999,5804,5804
 5804 IF(KZ2)4999,5805,4999
 5805 IF(ABS(U(L))+ABS(V(L))+ABS(W(L))+AIX(L))5806,4999,5806
 5806 KZ2=1
      GO TO 4999
 4999 IF(KZ1)5300,5222,5300
 5222 IF(K1-1)5300,5300,5000
 5000 IF(K1-K)5300,5001,5001
 5001 IF(ABS(U(L))+ABS(V(L))+ABS(W(L))+AIX(L))5002,5300,5002
 5002 KZ1=1
 5300 IF(JY1)5600,5304,5600
 5304 IF(J1-1)5600,5600,5301
 5301 IF(J1-J)5600,5302,5302
 5302 IF(ABS(U(L))+ABS(V(L))+ABS(W(L))+AIX(L))5303,5600,5303
 5303 JY1=1
 5600 IF(IX1)3342,5604,3342
 5604 IF(I1-1)3342,3342,5601
 5601 IF(I1-I)3342,5602,5602
 5602 IF(ABS(U(L))+ABS(V(L))+ABS(W(L))+AIX(L))5603,3342,5603
 5603 IX1=1
 3342 CONTINUE
  551 IF(AMX(L))9919,553,9980
 9980 IF(AMX(L)/(DX(I)*DY(J)*DZ(K))-TOZONE)9981,552,552
 9981 AMLOST=AMLOST+AMX(L)
      WS=U(L)**2+V(L)**2+W(L)**2
      WSR=AMX(L)*(AIX(L)+WS/2.)
      SUM=SUM+WSR
      ELOST=ELOST+WSR
      XMLOST=XMLOST+AMX(L)*U(L)
      YMLOST=YMLOST+AMX(L)*V(L)
      ZMLOST=ZMLOST+AMX(L)*W(L)
      AMX(L)=0.
  553 AIX(L)=0.
      U(L)=0.
      V(L)=0.
      W(L)=0.
      P(L)=0.
C
C     HERE THE FLUX DATA FROM THE  RIGHT  IS SET TO THE LEFT
  552 GAMC(J)=AMMP
      FLEFT(J)=AMUR
      YAMC(J)=AMVR
      ZMOM(J)=AMWR
      SIGC(J)=DELER
C
C     HERE THE FLUX DATA FROM THE  TOP    IS SET TO THE BOTTOM
C
  554 AMMY=AMPY
      AMMU=AMUT
      AMMV=AMVT
      AMMW=AMWT
      DELEB=DELET
C
```

```
C       HERE THE FLUX DATA FROM IN FRONT IS SET  TO THE BACK
  555 BMASS(M)=FMASS
      BYMOM(M)=FYMOM
      BXMOM(M)=FXMOM
      BZMOM(M)=FZMOM
      BENR(M)= FENR
      CONTINUE
      L=L+IMAX
C     $$$$$$$$$$$$$ END OF J LOOP $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    3 CONTINUE
 5700 LJ=L-IMAX
      IF(JY2)4,5701,4
 5701 IF(ABS(U(LJ))+ABS(V(LJ))+ABS(W(LJ))+AIX(LJ))5702,4,5702
 5702 JY2=1
    4 CONTINUE
  556 CONTINUE
C     $$$$$$$$$$$$$$ END OF I LOOP $$$$$$$$$$$$$$$$$$$
    2 CONTINUE
  557 CONTINUE
C     $$$$$$$$$$$$$$$$$$$$$$ END OF K LOOP $$$$$$$$$$$$$$$$$$$$$$$$$$$$
    1 CONTINUE
 8001 ETH=ETH-SUM
      ENEG=ENEG-SUM
      I1=I1-IX1
 5900 IF(I1)5901,5901,5902
 5901 I1=1
 5902 I2=I2+IX2
 5903 IF(I2-IMAX)5905,5905,5904
 5904 I2=IMAX
 5905 J1=J1-JY1
      IF(J1)5906,5906,5907
 5906 J1=1
 5907 J2=J2+JY2
      IF(J2-JMAX)5909,5909,5908
 5908 J2=JMAX
 5909 K1=K1-KZ1
      IF(K1)5910,5910,5911
 5910 K1=1
 5911 K2=K2+KZ2
      IF(K2-KMAX)5913,5913,5912
 5912 K2=KMAX
 5913 IF(REZ)9950,9950,9951
 9951 CALL REZONE
 9950 RETURN
 9903 NK=600
      GO TO 9999
 9904 NK=601
      GO TO 9999
 8999 NK=9
      GO TO 9999
 9900 NK=15
      GO TO 9999
 9901 NK=506
      GO TO 9999
 9902 NK=23
      GO TO 9999
 9910 NK=1103
```

```
      GO TO 9999
 9911 NK=106
      GO TO 9999
 9912 NK=317
      GO TO 9999
 9913 NK=702
      GO TO 9999
 9914 NK=705
      GO TO 9999
 9915 NK=7005
      GO TO 9999
 9916 NK=9982
      GO TO 9999
 9917 NK=7105
      GO TO 9999
 9918 NK=544
      GO TO 9999
 9919 NK=551
 9999 NR=4
      WRITE(6,8500)I,J,K,L,M,N,NN,NK,NR
      WRITE(6,8501)GAMC(J),FLEFT(J),YAMC(J),ZMOM(J),SIGC(J)
      WRITE(6,8501)AMMP,AMUR,AMVR,AMWR,DELER
      WRITE(6,8501)AMMY,AMMU,AMMV,AMMW,DELEB
      WRITE(6,8501)AMPY,AMUT,AMVT,AMWT,DELET
      WRITE(6,8501)BMASS(M),BXMOM(M),BYMOM(M),BZMOM(M),BENR(M)
      WRITE(6,8501)FMASS,FXMOM,FYMOM,FZMOM,FENR
      WRITE(6,8501)AMX(L),AIX(L),U(L),V(L),W(L),P(L),CYCLE
      WRITE(6,8501)AMX(L+1),U(L+1),V(L+1),W(L+1)
      WRITE(6,8501)AMX(NN),U(NN),V(NN),W (NN)
      WRITE(6,8501)AMX(L-1),U(L-1),V(L-1),W(L-1)
      WRITE(6,8501)AMX(N),U(N),V(N),W(N)
      LL=L-IMAX
      LBJ=L-IXMAX
      WRITE(6,8501)AMX(LL),U(LL),V(LL),W(LL)
      WRITE(6,8501)AMX(LBJ),U(LBJ),V(LBJ),W(LBJ)
 8501 FORMAT(1P8E12.5)
 8500 FORMAT(9I6)
      CALL UNCLE
      CALL DUMP
      END
@I FOR   REZONE/S,REZONE/S,REZONE/SS
      SUBROUTINE REZONE
C       CHANGE ALL CELL DIMENSIONS BY A FACTOR OF 2.
C         NOTE, 8 CELLS BECOME ONE IN THE  NEW GRID
C     CALCULATE NEW INDICES
      KKMAX=KMAX/2
      IIMAX=IMAX/2
      JJMAX=JMAX/2
C     SET UP DO LOOP FOR NEW STORAGE
      KN1=-IXMAX
      DO 21 KKK=1,KMAX,2
      LL=(KKK-1)*IXMAX
      KN1=KN1+IXMAX
      KN=KN1
      DO 21 II=1,IMAX,2
      I=LL+II
      KN=KN+1
```

```
      KK=KN
      DO 21 JJ=1,JMAX,2
      K=I+IXMAX
      J=I+IMAX
      L=K+IMAX
      WSA=AMX(I)+AMX(I+1)+AMX(J)+AMX(J+1)+
     1AMX(K)+AMX(K+1)+AMX(L)+AMX(L+1)
      IF(WSA)6,6,3
C     CALCULATE TWICE THE KINETIC ENERGY
    3 WSB=AMX(I)*(U(I)**2+V(I)**2+W(I)**2)+
     1AMX(I+1)*(U(I+1)**2+V(I+1)**2+W(I+1)**2)+
     2AMX(J)*(U(J)**2+V(J)**2+W(J)**2)+AMX(J+1)*(U(J+1)**2
     3+V(J+1)**2+W(J+1)**2)
      WSB=WSB+AMX(K)*(U(K)**2+V(K)**2+W(K)**2)+AMX(K+1)*
     1(U(K+1)**2+V(K+1)**2+W(K+1)**2)+AMX(L)*(U(L)**2+V(L)**2
     2+W(L)**2)+AMX(L+1)*(U(L+1)**2+V(L+1)**2+W(L+1)**2)
C     CALCULATE THE NEW VELOCITIES
    4 U(KK)=(AMX(I)*U(I)+AMX(I+1)*U(I+1)+AMX(J)*U(J)+AMX(J+1)*U(J+1)+
     1AMX(K)*U(K)+AMX(K+1)*U(K+1)+AMX(L)*U(L)+AMX(L+1)*U(L+1))/WSA
      V(KK)=(AMX(I)*V(I)+AMX(I+1)*V(I+1)+AMX(J)*V(J)+AMX(J+1)*V(J+1)+
     1AMX(K)*V(K)+AMX(K+1)*V(K+1)+AMX(L)*V(L)+AMX(L+1)*V(L+1))/WSA
      W(KK)=(AMX(I)*W(I)+AMX(I+1)*W(I+1)+AMX(J)*W(J)+AMX(J+1)*W(J+1)+
     1AMX(K)*W(K)+AMX(K+1)*W(K+1)+AMX(L)*W(L)+AMX(L+1)*W(L+1))/WSA
C     CALCULATE THE TOTAL INTERNAL ENERGY
      WSC=AMX(I)*AIX(I)+AMX(I+1)*AIX(I+1)+AMX(J)*AIX(J)+
     1AMX(J+1)*AIX(J+1)+AMX(L)*AIX(L)+AMX(L+1)*AIX(L+1)+AMX(K)*
     2AIX(K)+AMX(K+1)*AIX(K+1)
      P(KK)=0.
C     SET THE NEW MASSES
      AMX(KK)=WSA
C     CALCULATE THE NEW KINETIC ENERGY (ACTUALLY
C     TWICE)
      WS=U(KK)**2+V(KK)**2+W(KK)**2
      E=WSC+WSB/2.
C     THE NEW SPECIFIC INTERNAL ENERGY IS THE
C     TOTAL LESS THE KINETIC
      AIX(KK)=E/WSA-.5*WS
      AMX(J)=0.
      AMX(J+1)=0.
      AMX(K)=0.
      AMX(K+1)=0.
      AMX(L)=0.
      AMX(L+1)=0.
      AMX(I+1)=0.
      U(J)=0.
      U(J+1)=0.
      U(K)=0.
      U(K+1)=0.
      U(L)=0.
      U(L+1)=0.
      U(I+1)=0.
      V(J)=0.
      V(J+1)=0.
      V(K)=0.
      V(K+1)=0.
      V(L)=0.
      V(L+1)=0.
```

```
      V(I+1)=0.
      W(J)=0.
      W(J+1)=0.
      W(K)=0.
      W(K+1)=0.
      W(L)=0.
      W(L+1)=0.
      W(I+1)=0.
      AIX(I+1)=0.
      AIX(J)=0.
      AIX(J+1)=0.
      AIX(K)=).
      AIX(K+1 '=0.
      AIX(L)=0.
      AIX(L+1)=0.
      IF(II-1)380,380,390
  380 IF(JJ-1)381,381,390
  381 IF(KK-1)7,7,390
  390 AMX(I)=0.
      U(I)=0.
      V(I)=0.
      W(I)=0
      P(I)=0.
      AIX(I)=0.
      GO TO 7
C     CAME HERE BECAUSE OF ZERO MASS
    6 AMX(KK)=0.
      U(KK)=0.
      V(KK)=0.
      W(KK)=0.
      P(KK)=0.
      AIX(KK)=0.
      P(KK)=0.
    7 KK=KK+IMAX
   21 I=I+2*IMAX
C     CHANGE ALL CELL DIMENSIONS
      WS=0.
      DO 10 I=1,IIMAX
      DX(I)=2.0*DX(I)
      X(I)=WS+DX(I)
      WS=X(I)
   10 CONTINUE
      II=IIMAX+1
      WS=X(IIMAX)
      DO 11 I=II,IMAX
      DX(I)=DX(IIMAX)
      X(I)=WS+DX(I)
      WS=X(I)
   11 CONTINUE
      WS=0.
      DO 13 J=1,JJMAX
      DY(J)=2.0*DY(J)
      Y(J)=WS+DY(J)
      WS=Y(J)
   13 CONTINUE
      JJ=JJMAX+1
      WS=Y(JJMAX)
```

```
      DO 14 J=JJ,JMAX
      DY(J)=DY(JJMAX)
      Y(J)=WS+DY(J)
      WS=Y(J)
   14 CONTINUE
      WS=0.
      DO 16 K=1,KKMAX
      DZ(K)=2.0*DZ(K)
      ZCOR(K)=WS+DZ(K)
      WS=ZCOR(K)
   16 CONTINUE
      KK=KKMAX+1
      WS=ZCOR(KKMAX)
      DO 17 K=KK,KMAX
      DZ(K)=DZ(KKMAX)
      ZCOR(K)=WS+DZ(K)
      WS=ZCOR(K)
   17 CONTINUE
      KK=KKMAX+1
      DO 30 K=KK,KMAX
      LL=(K-1)*IXMAX
      DO 30 I=1,IMAX
      L=LL+I
      DO 30 J=1,JMAX
      AMX(L)=0.
      U(L)=0.
      V(L)=0.
      W(L)=0.
      AIX(L)=0.
      P(L)=0.
   30 L=L+IMAX
      DO 100 K=1,KKMAX
      LL=(K-1)*IXMAX
      DO 100 I=1,IIMAX
      L=LL+IIMAX+1-I
      M=LL+IMAX-I4+1-I
      DO 100 J=1,JJMAX
      AMX(M)=AMX(L)
      U(M)=U(L)
      V(M)=V(L)
      W(M)=W(L)
      AIX(M)=AIX(L)
      P(M)=P(L)
      AMX(L)=0.
      U(L)=0.
      V(L)=0.
      W(L)=0.
      AIX(L)=0.
      P(L)=0.
      M=M+IMAX
      L=L+IMAX
  100 CONTINUE
      I3=I3/2
C     NOW BEGIN ADDING ON MASS IN FRONT, BOTH
C      SIDES AND ABOVE.
      II=IIMAX-I4
      JJ=I3+1
```

```
C      NOTE, I4=NO. OF ZONES TO THE RIGHT TO ADD.
C     I3= INITIAL INTRFACE BETWEEN PROJECTILE AND TARGET.
      DO 200 K=1,KMAX
      LL=(K-1)*IXMAX
      DO 200 I=1,II
      L=LL+I+I3*IMAX
      DO 200 J=JJ,JMAX
      AMX(L)=DX(I)*DY(J)*DZ(K)*RHONOT
  200 L=L+IMAX
      II=IMAX-I4
      IL=IIMAX-I4+1
      JL=JJMAX+1
      DO 300 K=1,KMAX
      LL=(K-1)*IXMAX
      DO 300 I=IL,II
      L=LL+I+JJMAX*IMAX
      DO 300 J=JL,JMAX
      AMX(L)=DX(I)*DY(J)*DZ(K)*RHONOT
  300 L=L+IMAX
      II=IMAX-I4+1
      JJ=I3+1
      DO 400 K=1,KMAX
      LL=(K-1)*IXMAX
      DO 400 I=II,IMAX
      L=LL+I+I3*IMAX
      DO 400 J=JJ,JMAX
      AMX(L)=DX(I)*DY(J)*DZ(K)*RHONOT
  400 L=L+IMAX
      II=IMAX-I4
      IL=IIMAX-I4+1
      JJ=I3+1
      IF(JJ-JJMAX)700,700,800
  700 KL=KKMAX+1
      DO 500 K=KL,KMAX
      LL=(K-1)*IXMAX
      DO 500 I=IL,II
      L=LL+I+I3*IMAX
      DO 500 J=JJ,JMAX
      AMX(L)=DX(I)*DY(J)*DZ(K)*RHONOT
  500 L=L+IMAX
  800 WS=T+DTNA
C       RESET ACTIVE GRID COUNTERS
      I1=I4-1
      I2=IMAX-I4+2
      J1=J1
      J2=JJMAX+2
      K1=K1
      K2=KKMAX+2
      NK=NC+1
      WRITE(6,9000)WS,NK
 9000 FORMAT(1H ////22H PROBLEM REZONED AT T=1PE12.6,6X,5HCYCLEI4////)
      WRITE(6,8000)IMAX,(X(I),I=1,IMAX)
      WRITE(6,8003)IMAX,(DX(I),I=1,IMAX)
      WRITE(6,8001)JMAX,(Y(J),J=1,JMAX)
      WRITE(6,8004)JMAX,(DY(J),J=1,JMAX)
      WRITE(6,8002)KMAX,(ZCOR(K),K=1,KMAX)
      WRITE(6,8005)KMAX,(DZ(K),K=1,KMAX)
```

```
      WRITE(6,8006)IMAX,JMAX,KMAX,IXMAX,KMAXA
      RETURN
 8000 FORMAT(1H /10H X(I) I=1,I2/(5F16.6))
 8001 FORMAT(1H /10H Y(J) J=1,I2/(5F16.6))
 8002 FORMAT(1H /13H ZCOR(K) K=1,I2/(5F16.6))
 8003 FORMAT(1H /11H DX(I) I=1,I2/(5F16.6))
 8004 FORMAT(1H /11H DY(J) J=1,I2/(5F16.6))
 8005 FORMAT(1H /11H DZ(K) K=1,I2/(5F16.6))
 8006 FORMAT(7I8)
      END
@I FOR   ES/S,ES/S,ES/SS
      SUBROUTINE ES                                                     ES  0010
C                                                                       PH2 0690
C
C     ********* 3DOIL ***************************
C
C       $$$$ FOR COMPLETE DETAILS, SEE  GA-3216,
C        METALLIC EQUATIONS OF STATE FOR HYPERVELOCITY IMPACT
C            BY JAMES TILLOTSON
C          IF THE MATERIAL IS COMPRESSED , USE THE CONDENSED
C          FORM OF THE EQUATION OF STATE.
C              IF THE MATERIAL IS RAREFIED AND IF THE SPECIFIC
C          INTERNAL ENERGY IS GREATER THAN E SUB S , USE
C         THE RARFIED FORM ... BUT---------
C         IF RAREFIED AND E IS LESS THAN E SUB S,
C           USE THE CONDENSED FORM.
C         $$$$ NOTE, NO NEGATIVE(TENSION) PRESSURES ALLOWED
C
C
C                                                                       ES  0980
   10 RHOW=AMX(L)/(DX(I)*DY(J)*DZ(K))
      ETA=RHOW/Z(33)
      VOW=1.0/ETA                                                       ES  1010
   11 P1=AIX(L)*RHOW*Z(34)
   12 P2=AIX(L)
   13 P3=Z(35)*ETA*ETA
   14 P4=Z(36)/(P2/P3+1.0)*AIX(L)*RHOW
   15 P5=Z(37)*(ETA-1.)
   16 IF(ETA-1.0)50,100,100                                             ES  1070
   50 IF(VOW-Z(38))55,55,75
   55 IF(AIX(L)-Z(40))100,100,75
   75 P7=Z(41)*(VOW-1.)
      IF(P7-88.0)4002,4002,4003                                         ES  1110
 4003 P7=88.0                                                           ES  1120
 4002 CONTINUE                                                          ES  1130
      P8=EXP(P7)                                                        ES  1140
      P9=1.0/P8                                                         ES  1150
      P10=Z(42)*((VOW-1.)**2)
      IF(P10-88.0)4000,4000,4001                                        ES  1170
 4001 P10=88.0                                                          ES  1180
 4000 CONTINUE                                                          ES  1190
      P11=EXP(P10)                                                      ES  1200
      P12=1.0/P11                                                       ES  1210
      P(L)=P1+(P4+P5*P9)*P12
      GO TO 119
  100 P6=Z(44)*((ETA-1.)**2)
      P(L)=P1+P4+P5+P6
```

```
  119 IF(P(L))999,999,200
  200 WSGX=.5                                                          ES   1260
      GO TO 500                                                        ES   1270
  999 P(L)=0.
      WSGX=.5+Z(43)
      GO TO 500                                                        ES   1300
  500 RETURN
      END
@I FOR   EDIT/S,EDIT/S,EDIT/SS
      SUBROUTINE EDIT                                                  EDIT0010
C                                                                      EDIT0990
C     ********* 3DOIL ***************************
C     HERE WE WILL DECIDE  WHETHER TO HAVE A SHORT PRINT, LONG PRINT,
C     DUMP ON THE BINARY TAPE OR STOP THE PROBLEM
C
  104 CALL SLITET(3,K000FX)                                            EDIT1040
       GO TO(106,108),K000FX                                           EDIT1050
  106 CALL SLITE (3)                                                   EDIT1060
      GO TO 126                                                        EDIT1070
  108 IF(CYCLE-CSTOP)110,122,122                                       EDIT1080
  110 IF(REZ)9901,112,124
  112 IF(AMOD(CYCLE,DUMPT7))114,124,114                                EDIT1100
  114 IF(AMOD(CYCLE,PRINTL))120,126,120
  120 IF(AMOD(CYCLE,PRINTS))140,128,140                                EDIT1150
  122 CALL SLITE (1)                                                   EDIT1160
  124 GO TO 1                                                          EDIT1170
  126 CALL SLITE (4)                                                   EDIT1180
  128 GO TO 6000                                                       EDIT1190
  130 GO TO 1000                                                       EDIT1200
  132 CALL SLITET(4,K000FX)                                            EDIT1210
       GO TO(134,136),K000FX                                           EDIT1220
  134 GO TO 5000                                                       EDIT1230
C
C     CALCULATE THE ENERGY CHECK = (ETH-E)/ETH AT CYCLE N+1 -
C     (ETH-E)/ETH AT CYCLE M  ALL DIVIDED  THRU  BY THE  NUMBER OF CYCLE
C           BETWEEN  ENERGY CHECKS =(CYCLE(N+1) -CYCLE M)
C
  136 IF(ABS(ECK)-DMIN)140,140,9905                                    EDIT1240
  140 CALL SLITET(1,K000FX)                                            EDIT1250
       GO TO(142,144),K000FX                                           EDIT1260
  142 REWIND(N7)
      CALL SLITE (1)                                                   EDIT1280
  144 GO TO 10000                                                      EDIT1290
    1 IF(DUMPT7)30,3,3                                                 EDIT1330
C
C     **** DUMP ALL THE CELL-CENTERED QUANTITIES ,THE X,S AND Y,S
C           AND ZCOR,S AND THE ENTIRE Z BLOCK
C
    3 BACKSPACE N7
      WS=555.0                                                         EDIT1360
      WRITE(N7)WS,CYCLE,N3
      WRITE(N7)(Z(L),L=1,MZ)
      WRITE(N7)(U(K),V(K),W(K),AMX(K),AIX(K),K=1,KMAXA)
      WRITE(N7)(X(I),I=1,IMAX)
      WRITE(N7)(Y(J),J=1,JMAX)
      WRITE(N7)(ZCOR(K),K=1,KMAX)
      WS=666.0                                                         EDIT1480
```

```
      WRITE(N7)WS,WS,WS
      WRITE(6,8120)NC
   30 GO TO 126                                                          EDIT1510
 6000 CONTINUE
      DO 6012 I=1,18
 6012 PR(I)=0.
C
C     CALCULATE  THE  TOTAL INTERNAL AND KINETIC ENERGY ,AND THE
C      TOTAL MASS.
C
      DO 6028 K=1,KMAXA
 6017 PR(1)=0.0                                                          EDIT1760
      PR(2)=0.0                                                          EDIT1770
      PR(4)=0.
 6019 IF(AMX(K))9917,6028,6020                                           EDIT1790
 6020 WSB=(U(K)**2+V(K)**2+W(K)**2)*.5
      PR(5)=PR(5)+AIX(K)*AMX(K)                                          EDIT1910
      PR(6)=PR(6)+WSB*AMX(K)                                             EDIT1920
      PR(8)=PR(8)+AMX(K)                                                 EDIT1930
 6028 CONTINUE                                                           EDIT1940
      PR(3)=PR(1)+PR(2)                                                  EDIT1950
      PR(7)=PR(5)+PR(6)                                                  EDIT1960
      XNRG=PR(7)                                                         EDIT1970
      PR(9)=PR(1)+PR(5)                                                  EDIT1980
      PR(10)=PR(2)+PR(6)                                                 EDIT1990
      PR(11)=PR(3)+PR(7)                                                 EDIT2000
      PR(12)=PR(4)+PR(8)                                                 EDIT2010
      WSA=(ETH-PR(11))/ETH                                               EDIT2020
      IF(CYCLE)9931,9931,9932
 9931 NPC=1
 9932 PR(18)=(WSA-DNN)/FLOAT(NPC)
      ECK=PR(18)                                                         EDIT2040
      DNN=WSA                                                            EDIT2050
      NPC=0                                                              EDIT2060
      SUM=0.0                                                            EDIT2070
C     RADEB= TOTAL POSITIVE Z MOM.
C     RADER = TOTAL POSITIVE X MOM.
C     RADET= TOTAL POSITIVE  Y MOM.
      SUMB=0.
      SUMR=0.
      SUMT=0.
      DO 810 K=1,KMAXA
      IF(AMX(K))810,810,802
  802 IF(V(K))804,804,803
  803 SUMT=SUMT+AMX(K)*V(K)
  804 IF(U(K))805,805,806
  806 SUMR=SUMR+AMX(K)*U(K)
  805 IF(W(K))810,810,808
  808 SUMB=SUMB+AMX(K)*W(K)
  810 CONTINUE
      RADEB=SUMB
      RADER=SUMR
      RADET=SUMT
      WRITE(6,8116)PROB,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
      WRITE(6,8117)(PR(I),I=1,8)
      WRITE(6,8118)(PR(I),I=9,12)
      WRITE(6,8119)RADEB,RADER,RADET,UVMAX,ETH,ECK
```

```
      WRITE(6,9040)N10,N11,N9,I1,I2,J1,J2,K1,K2
      WRITE(6,9042)AMLOST,ELOST,XMLOST,YMLOST,ZMLOST,ENEG
 9042 FORMAT(1P7E14.7)
 6090 GO TO 130                                                    EDIT2410
 1000 GO TO 1030                                                   EDIT2460
 1030 WRITE(6,8116)PROB,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
      JMAX=JMAX                                                    EDIT2480
      WRITE(6,8307)X1,X2,XMAX,Y1,Y2,Y(JMAX)
      GO TO 132
 5000 WRITE(6,8116)PROB,NC,T,DTNA,TRAD,DTRAD,NR,N1,N2,N3,N4
C     NOTICE THE LIMITS OF THE DO LOOPS
      DO 1126 KK=K1,K2
C
C     HERE WE  PREPARE FOR THE LONG PRINT
C
      WRITE(6,9041)KK,ZCOR(KK),DZ(KK)
 5004 DO 5050 I=I1,I2
      CALL SLITE (4)                                               EDIT2930
      J=J2+1
      K=J2*IMAX+I+(KK-1)*IXMAX
      DO 5046 L=J1,J2
      J=J-1                                                        EDIT2970
      K=K-IMAX                                                     EDIT2980
 5012 IF(AMX(K))9917,5046,5014                                     EDIT2990
 5014 CALL SLITET(4,K000FX)                                        EDIT3000
       GO TO(5016,5019),K000FX                                     EDIT3010
 5016 WRITE(6,8135)I,X(I),DX(I)
 5019 WS=AMX(K)/(DX(I)*DY(J)*DZ(KK))
      WSC=P(K)*1.E+4                                               EDIT3050
C
C     J=ROW NUMBER
C
C     U= X COMPONENT OF VELOCITY  CM./SH.
C
C     V= Y COMPONENT OF VELOCITY  CM./SH.
C
C     WSC = PRESSURE IN MEGABARS
C
C     AMX = MASS IN GRAMS
C
C     WS= DENSITY IN GRAMS/CM CUBED
C
C     W= Z COMPONENT OF VELOCITY  CM./SH.
C
C     AIX= SPECIFIC INTERNAL ENERGY IN  JERKS/GM. (1 JERK= 10(16)ERGS)
C
C     Y= TOP COORDINATE OF THIS ROW  IN CM.
C
C
 5018 WRITE(6,8108)J,U(K),V(K),WSC,AMX(K),WS,AIX(K),W(K),Y(J)
 5046 CONTINUE                                                     EDIT3080
 5050 CONTINUE                                                     EDIT3090
 1126 CONTINUE
      GO TO 136                                                    EDIT3100
 9901 NK=110                                                       EDIT3150
      GO TO 9999                                                   EDIT3160
 9905 NK=136                                                       EDIT3180
```

```
      GO TO 9999                                                        EDIT3190
 9917 NK=6015                                                           EDIT3210
      GO TO 9999                                                        EDIT3220
 9999 NR=6                                                              EDIT3320
      WRITE(6,8002)I,J,K,KP,I1,I2,NK,NR
 8002 FORMAT(8I5)
      CALL UNCLE
      CALL DUMP
10000 RETURN                                                            EDIT3340
C                                                                       EDIT3350
C                    FORMATS                                            EDIT3360
 8108 FORMAT(I3,1X,1P8E12.5)
 81160FORMAT(8H1PROBLEM6X,5HCYCLE9X,4HTIME13X,2HDT13X,4HTRAD11X,5HDTRAD1EDIT3380
     12X,2HNR6X,2HN14X,2HN24X,2HN34X,2HN4/(F7.1,I11,2X,1P4E16.7,I10,2X,4EDIT3390
     2I6))                                                              EDIT3400
 81170FORMAT(1H0//17X2HAI16X,2HAK14X,5HAI+AK15X,2HAM/4H DOT3X,1P4E18.7/3EDIT3410
     1H  X4X,4E18.7)                                                    EDIT3420
 81180FORMAT(12X,13H-------------5X,13H-------------5X,13H-------------5EDIT3430
     1X,13H-------------/7H TOTALS1P4E18.7)                             EDIT3440
 81190FORMAT(2H0 //16X,5HRADEB13X,5HRADER13X,5HRADET12X,7HMAX VEL13X,3HTEDIT3450
     1HE12X,9HREL ERROR/7X,1P6E18.7////)                                EDIT3460
 8120 FORMAT(1H0//21H TAPE 7 DUMP ON CYCLEI5////)                       EDIT3470
 81350FORMAT(1H ///4H I =I3,6X,6HX(I) =F12.3,6X,7HDX(I) =F12.3//3H   J8X,EDIT3520
     11HX10X,1HY10X,3HF/A9X,3HAMX9X,3HRHO8X,3HAIX9X,4H   W 8X,2H Y/)
 8307 FORMAT(5H X1 =1PE12.5,3X,4HX2 =E12.5,3X,6HXMAX =E12.5,6X,4HY1 =E12
     1.5,3X,4HY2 =E12.5,3X,6HYMAX =E12.5)
 9040 FORMAT(1H / 9I6)
 9041 FORMAT(1H ///4H K =I3,6X,9HZCOR(K) =F12.3,6X,7HDZ(K) =F12.3)
      END                                                               EDIT3630
```

5. INPUT AND DEFINITIONS OF THE VARIABLES

5.1 Normal Input for the TRIOIL Code

An asterisk (*) implies that the data on the card is to be converted to fixed point data (requires a 2 punch in Column 1). All data loaded via the card routine is read by a floating point format. A double asterisk (**) signifies that this is the last data card in this set, and requires a one in Column 1.

First Set

The number of BCD (header cards) that will be read in (Columns 1-3, format I3).

N BCD cards, alphameric and or numeric in Columns 2-72.

Second Set

Location	Symbol	Description
*103	N7	Binary tape number (data tape)
**36271	PK(1)	Problem number
36272	PK(2)	The cycle number to start the calculation
36273	PK(3)	If < 0. code assumes that this is a re-start or that the CLAM code has generated the initial data. (NOTE: At this time of writing the report, a three-dimensional version of the CLAM code is not available.) If PK(3) ≥, the code will call subroutine set-up.

Third Set

Location	Symbol	Description
1	PROB	Problem number, identical to the value in PK(1)
3	DT	The time step, Δt^n in shakes. (NOTE: 1 shake = 10^{-8} seconds)
4	PRINTS	Short print frequency in cycles
5	PRINTL	Long print frequency in cycles
6	DUMPT7	Binary tape dump frequency in cycles

Location	Symbol	Description
7	CSTOP	Cycle at which the problem will stop
13	FFA	Upper limit for stability and to calculate Δt if CABLN = 0.
14	FFB	Lower limit for stability and to calculate Δt if CABLN = 0.
20	DMIN	If ECK (Z(24)) > DMIN, problem will stop because of poor energy consertation
23	TOZONE	$\sim 10^{-4}\ \rho_o$ = minimum density for mass transport at a free surface within the grid
25	SBOUND	= 1.0, fraction of Δ in. the weighted velocity term in the calculation of the mass flux
26	CABLN	If < 0. the code will control Δt at PCSTAB (an input number) of stability If = 0. the code will control Δt, decreasing Δt if $\lvert u$ or $c\rvert\ \Delta t/\Delta x$ or $\lvert v$ or $c\rvert\ \Delta t/\Delta y$ or $\lvert w$ or $c\rvert\ \Delta t/\Delta z$ exceed FFA (an input number) and increasing Δt if the above terms are less than FFB (an input number) If > 0., the Δt that is loaded at t = 0., will remain constant, regardless of any stability considerations
	NOTE: These 2 options require that you load Δt (location 3) at time t = 0.	
29	WSGD	= GAMMA for the dot material
30	WSGX	= GAMMA for the x material
45	DTCHK	$\sim 10^{-4}\ \rho_o$, any cell that has a density less than this value, will be bypassed for stability checks
46	PCSTAB	$\sim$ .25, fraction of stability as determined by the Courant condition or particle velocity
58	Z(58)	Initial x velocity component of the projectile in cm/shake
59	Z(59)	Initial z velocity component of the projectile in cm/shake
66	RHONOT	Initial density (ρ_o) of all material (since this is a one-material code)
67	VELOC	Initial y velocity componenet of the projectile in cm/shake
68	BUG	$\sim$ .05, epsilon for determining whether special features will be used to empty the bottom cells in the projectile

Location	Symbol	Description
69	Z(69)	= percent of (PCSTAB) at time t = 0. used for problems in which most of the energy at t = 0. is in the form of internal energy
70	Z(70)	Factor to increase Z(69) every cycle up until Z(69) is equal to 1.0
*86	iMAX	Maximum number of zones in the x direction
*87	jMAX	Maximum number of zones in the y direction
*88	kMAX	Maximum number of zones in the z direction
75	Z(75)	Density of material leaving the left boundary of grid in order to trigger rezone
76	Z(76)	Similar term for the front boundary of grid
77	Z(77)	Similar term for the right boundary of grid
78	Z(78)	Similar term for the back boundary of grid
79	Z(79)	Similar term for the top boundary of grid
80	Z(80)	Similar term for the bottom boundary of grid
*95	i3	The original (j) value of projectile-target interface
*96	i4	The number of zones to add on to the right side of grid after rezone. i4 + the number to add on the left is = to iMAX/2
*97	N1	If = 0. the left side of grid is transmittive, otherwise reflective
*98	N2	Similar term for the right side of grid
*99	N3	Similar term for the top side of grid
*100	N4	Similar term for the bottom side of grid
*101	N5	Similar term for the back side of grid
*102	N6	Similar term for the front side of grid
*103	N7	Binary tape number
36151	Dx	= Δx to be used for all (i)
36181	Dy	= Δy to be used for all (j)
36211	Dz	= Δz to be used for all (k)
50	S1	= interface (j) value + 1, between the projectile and the target
51	S2	= back (k) boundary + 1 of the projectile
52	S3	= front (k) boundary of the projectile
53	S4	= the left (i) boundary + 1, of the projectile

Location	Symbol	Description
54	S5	= the right (i) boundary of the projectile
55	S6	= the bottom (j) boundary + 1, of the projectile
56	S7	= the top boundary of the projectile (j)
*93 *94	i1 i2	= minimum and maximum values of the active grid in the x direction
*108 *109	k1 k2	= minimum and maximum values of the active grid in the z direction
*110 *111	j1 j2	= minimum and maximum values of the active grid in the y direction
47 48 49	Z(47) Z(48) Z(49)	$= c_o$ $= A_1$ $= A_2$ $c = c_o + A_1(P)^{A2}$ Where P is in megabars and c_o and A_1 are in units of 10^5 cm/sec, converted to cm/shake in CDT routine
33 34 35 36 37 38 39 40 41 42 43 44	Z(33) Z(34) Z(35) Z(36) Z(37) Z(38) Z(39) Z(40) Z(41) Z(42) Z(43) Z(44)	$= \rho_o$ $= a$ $= E_o$ $= b$ $= A$ $= V_s$ Blank $= E_s$ $= \alpha$ $= \beta$ Blank $= B$ Constant for metallic equation of state (Tillotson formulation)
16	xMAX	= epsilonics on the velocity, if $\|u\|$ or $\|v\|$ or $\|w\| < \epsilon$, it is set to 0. and the books are kept
15	TMASS	= epsilonics on the specific internal energy. If $I < \epsilon$, it is set to 0. and the books are kept
**22	REZFCT	No meaning in this code

Last Set

Location	Symbol	Description
**22	REZFCT	No meaning

5.2 List of Common for TRIOIL

Symbol	Location	No. of Words	Units	Description
AIX	151	6000	jerks/g	Specific internal energy for cell (L)
AMX	6151	6000	g	Mass in cell (L)
U	12151	6000	cm/shake	X component of velocity in cell (L)
V	18151	6000	cm/shake	Y component of velocity in cell (L)
W	24151	6000	cm/shake	Z component of velocity in cell (L)
P	30151	6000	jerks/cm^3	Material pressure in cell (L)
DX	36151	30	cm	DX(i) = X(i) - X(i-1)
DY	36181	30	cm	DY(j) = Y(j) - Y(j-1)
DZ	36211	30	cm	DZ(k) = ZCOR(k) - ZCOR(k-1)
UL	36241	30	cm/shake	Velocity at the left of cell in PH1
FLEFT	36241	30	g cm/shake	X momenta of mass crossing left side of cell (PH2)
PL	36271	30	jerks/cm^3	Temporary pressure array at the left interface in PH1
PK	36271	30	none	Not used (except input)
YAMC	36271	30	g cm/shake	Y momenta of mass crossing left side of cell PH2
X	36301	30	cm	X(i) = right dimension of cell (L)
Y	36331	30	cm	Y(j) = top dimension of cell (L)
ZCOR	36361	30	cm	ZCOR(k) = Z or front dimension of cell (L)
PR	36391	50	many	Used for editing in the EDIT routine
SIGC	36441	30	jerks/g	= specific energy of the mass crossing left side of cell (PH2)
GAMC	36471	30	g	Mass crossing left side of cell in (PH2)

Symbol	Location	No. of Words	Units	Description
ZMOM	36501	30	g cm/shake	Z momenta of mass crossing left isde of cell (PH2)
BXMOM	36531	700	g cm/shake	X momenta of mass crossing back side of cell (PH2)
UBIND	36531	700	cm/shake	Velocity at back interface of cell in PH1
PBIND	37231	700	jerks/cm^3	Pressure array at back interface of cell in (PH1)
BMASS	37231	700	g	Mass crossing back interface of cell in PH2
BYMOM	37931	700	g cm/shake	Y momenta of mass crossing back surface in PH2
BZMOM	38631	700	g cm/shake	Z momenta of mass crossing back surface in PH2
BENR	39331	700	jerks/g	Specific energy of this mass crossing the back surface in PH2
AREA	40031	1	none	Flag in PH2 for the Z direction
BIG	40032	1	none	Not used
BOUNCE	40033	1	none	Not used
PABOVE	40034	1	jerks/cm^3	Pressure at the top of cell (L) in PH1
PBLO	40035	1	jerks/cm^3	Pressure at the bottom of cell (L) in PH1
PiDTS	40036	1	none	Not used
PRR	40037	1	jerks/cm^3	Pressure at the right of cell (L) in PH1
RHO	40038	1	none	Not used
SIG	40039	1	cm	Minimum Δx, Δy or Δz for a cell in CDT routine
UVMAX	40040	1	shake^{-1}	(Maximum velocity)/minimum (Δx or Δy or Δz)
VABOVE	40041	1	cm/shake	Velocity at the top of cell (L) in PH1
VBLO	40042	1	cm/shake	Velocity at the bottom of cell (L) in PH1
VEL	40043	1	none	Maximum (α-1) in the CDT routine, a flag for subcycling in PH1 and a flag for the y direction in PH2

Symbol	Location	No. of Words	Units	Description
WPS	40044	1	none	Working storage
WS	40045	1	none	Working storage
WSA	40046	1	none	Working storage
WSB	40047	1	none	Working storage
WSC	40048	1	none	Working storage
i	40049	1	none	Temporary indices
ii	40050	1	none	Temporary indices
iN	40051	1	none	Temporary indices
iR	40052	1	none	Temporary indices
iWS	40053	1	none	Temporary indices
iWSA	40054	1	none	Temporary indices
iWSB	40055	1	none	Temporary indices
iWSC	40056	1	none	Temporary indices
j	40057	1	none	Temporary indices
jN	40058	1	none	Temporary indices
jP	40059	1	none	Temporary indices
jR	40060	1	none	Temporary indices
K	40061	1	none	Temporary indices
KDT	40062	1	none	Flag in CDT for Δt change
KN	40063	1	none	Temporary indices
KP	40064	1	none	Temporary indices
KR	40065	1	none	Temporary indices
LRM	40066	1	none	Temporary indices
L	40067	1	none	Temporary indices
M	40068	1	none	Temporary indices
MA	40069	1	none	Temporary indices
MB	40070	1	none	Temporary indices
MC	40071	1	none	Temporary indices
MD	40072	1	none	Temporary indices
ME	40073	1	none	Temporary indices
MZ	40074	1	none	Set = 150 in input required in EDIT
N	40075	1	none	Temporary indices

Symbol	Location	No. of Words	Units	Description
REZ	40076	1	none	Not used
TRAD	40077	1	none	Not used
DTRAD	40078	1	none	Not used
RADEB	40079	1	none	Total positive Z momentum
RADER	40080	1	none	Total positive X momentum
RADET	40081	1	none	Total positive Y momentum
X1	40082	1	none	Not used
X2	40083	1	none	Not used
Y1	40084	1	none	Not used
Y2	40085	1	none	Not used
iMAXA	40086	1	none	Not used

The Z Block

Location	Symbol	Units	Description
Z(1)	PROB	none	Problem number
Z(2)	CYCLE	none	Floating point value of the cycle number
Z(3)	DT	shake	Δt hydro = $t^n - t^{n-1}$
Z(4)	PRINTS	none	Cycle frequency for short print
Z(5)	PRINTL	none	Cycle frequency for long print
Z(6)	DUMPT7	none	Cycle frequency for binary tape dump
Z(7)	CSTOP	none	Cycle number at which problem stops
Z(8)	PIDY	none	= π = 3.1415927
Z(9)	GAM	none	Not used
Z(10)	GAMD	none	= $1./(\alpha-1.)$ computer in input
Z(11)	GAMX	none	= $1./(\alpha_x-1.)$ routine
Z(12)	ETH	jerks	= total energy in the system (originally set = to (iMAX)(jMAX)(kMAX)) $\sum_{L=1} AMX_{(L)}\left[I_{(L)}+\frac{1}{2}U^2_{(L)}+V^2_{(L)}+W^2_{(L)}\right]$ Changed to PH1 at transmittive boundaries and in PH2 if mass leaves the system

Location	Symbol	Units	Description
Z(13)	FFA	none	Upper limit for stability and used to calculate Δt only if CABLN = 0.
Z(14)	FFB	none	Lower limit for stability and used to calculate Δt only if CABLN = 0.
Z(15)	TMASS	none	Epsilonics on minimum specific internal energy
Z(16)	XMAX	none	Epsilonics on minimum velocity components
Z(17)	YMAX	none	Not used
Z(18)	ZMAX	none	Not used
Z(19)	DNN	none	= $(ETH-E^{N-NPC}/ETH)$ for energy check
Z(20)	DMIN	none	if ECK (see definition in Z(24) is > DMIN, problem will stop because of energy violation
Z(21)	DTNA	shake	Δt^{n-1}
Z(22)	REZFCT	none	Not used
Z(23)	TOZONE	g/cm^3	If the mass flow across a free surface within the grid produces a density < TOZONE, the mass flow is set to zero
Z(24)	ECK	none	Is the energy check $(ETH-E^n/ETH) - (ETH-E^{N-NPC}/ETH)/NPC$ Where NPC = cycle frequency at which the energy check is made
Z(25)	SBOUND	none	Fractions of Δ in mass weighted velocity (suggested number = 1.0)
Z(26)	CABLN	none	Defined in Section 5.1
Z(27)	T	shake	Total time up to cycle NC $t^{n+1} = t^n + \Delta t$
Z(28)	GMAX	none	Maximum of the two gammas (γ_x or γ.)
Z(29)	WSGD	none	γ.
Z(30)	WSGX	none	γ_x and $(\gamma_{MAX-1.})$ in CDT routine
Z(31)	GMADR	none	$\gamma/(\gamma-1.)$
Z(32)	GMAXP	none	$\gamma_x/(\gamma_x-1.)$

Location	Symbol	Units	Description
Z(33)	ρ_o	g/cm^3	For metallic equations of state (Tillotson formulation)
Z(34)	a	none	
Z(35)	E_o	jerks/g	
Z(36)	b	none	
Z(37)	A	$jerks/cm^3$	
Z(38)	V_s	none	
Z(39)	none	none	
Z(40	E_s	jerks/g	
Z(41)	α	none	
Z(42)	β	none	
Z(43)	O	none	
Z(44)	B	$jerks/cm^3$	
Z(45)	DTCHK	g/cm^3	Density checks, if $\rho_{(L)} <$ DTCHK, the stability check for cell (L) will be bypassed
Z(46)	PCSTAB	none	% of stability, used if CABLN is < 0. A recommended value is .25
Z(47)	CNOT	10^5cm/sec	C = speed of sound = CNOT + BFACT$(P_{(L)})^{EPSI}$ where $P_{(L)}$ is in megabars
Z(48)	BFACT	none	
Z(49)	EPSi	none	
Z(50)	S1	none	Used in set-up routine
Z(51)	S2	none	
Z(52)	S3	none	
Z(53)	S4	none	
Z(54)	S5	none	
Z(55)	S6	none	
Z(56)	S7	none	
Z(57)	S8	none	Not used
Z(58)	S9	none	Initial X velocity component of the projectile in cm/shake
Z(59)	S10	none	Similar term for the Z direction
Z(60)	AMLOST	g	Mass thrown away (PH2)
Z(61)	ELOST	jerks	Energy thrown away (PH2)
Z(62)	XMLOST	g cm/shake	Total X momenta thrown away (PH2)
Z(63)	YMLOST	g cm/shake	Total Y momenta thrown away (PH2)
Z(64)	ZMLOST	g cm/shake	Total Z momenta thrown away (PH2)
Z(65)	ENEG	jerks	Energy added to system if I < 0.
Z(66)	RHONOT	g/cm^3	Initial density of material
Z(67)	VELOC	cm/shake	Initial velocity of pellet in the Y direction (cm/shake)
Z(68)	BUG	none	Epsilon for emptying pellet (~ .01)

Location	Symbol	Units	Description
Z(69)		none	Defined in Section 5.1
Z(70)		none	Defined in Section 5.1
Z(71)		none	j value of top of target
Z(72)		none	Not used
Z(73)		none	Not used
Z(74)		none	Not used
Z(75)		none	Defined in Section 5.1
Z(76)		none	Defined in Section 5.1
Z(77)		none	Defined in Section 5.1
Z(78)		none	Defined in Section 5.1
Z(79)		none	Defined in Section 5.1
Z(80)		none	Defined in Section 5.1
Z(81)	NPR	none	Index (working storage)
Z(82)	NPRi	none	Index (working storage)
Z(83)	NC	none	Cycle number in fixed point
Z(84)	NPC	none	Number or cycles between energy checks
Z(85)	NRC	none	Index
Z(86)	iMAX	none	Maximum number of zones in X direction
Z(87)	jMAX	none	Maximum number of zones in Y direction
Z(88)	kMAX	none	Maximum number of zones in Z direction
Z(89)	kMAXA	none	Total number of zones = (iMAX)(jMAX)(kMAX)
Z(90)	iXMAX	none	= (iMAX)(jMAX)
Z(91)	NOD	none	Index (working storage)
Z(92)	NOPR	none	Index (working storage)
Z(93)	i1	none	Minimum value of i in do loop on i
Z(94)	i2	none	Maximum value of i in do loop on i
Z(95)	i3	none	Original interface between projectile and target

Location	Symbol	Units	Description
Z(96)	i4	none	Number of zones to the right for rezone
Z(97)	N1	none	Flag at left, if 0., boundary is transmittive, otherwise reflective
Z(98)	N2	none	Flag at right, if 0., boundary is transmittive, otherwise reflective
Z(99)	N3	none	Flag at top, if 0., boundary is transmittive, otherwise reflective
Z(100)	N4	none	Flag at bottom, if 0., boundary is transmittive, otherwise reflective
Z(101)	N5	none	Flag behind, if 0., boundary is transmittive, otherwise reflective
Z(102)	N6	none	Flag in front, if 0., boundary is transmittive, otherwise reflective
Z(103)	N7	none	Binary tape number designation
Z(104)	N8	none	Not used
Z(105)	N9	none	k value of zone that is controlling Δt
Z(106)	N10	none	i value of zone that is controlling Δt
Z(107)	N11	none	j value of zone that is controlling Δt
Z(108)	k1	none	Minimum value of k in do loop on k
Z(109)	k2	none	Maximum value of k in do loop on k
Z(110)	j1	none	Minimum value of j in do loop on j
Z(111)	j2	none	Maximum value of j in do loop on j
Z(112) through Z(150)			Not used

REFERENCES

1. Johnson, W. E., "OIL, A Continuous Two-Dimensional Eulerian Hydrodynamic Code," GAMD-5580. January 7, 1965 (Rev.)

2. Dienes, J. K., W. E. Johnson and J. M. Walsh, "Annual Status Report on the Theory of Hypervelocity Impact," GA-6509. June 28, 1965.

3. Harlow, F. H., "Two-Dimensional Hydrodynamic Calculations," LA-2301. September 1959.

4. Johnson, W. E., "TOIL (A Two-Material Version of the OIL Code)," GAMD-8073. July 13, 1967.